RETURN TO

AEDC-TR-72-162

AERODYNAMIC FORCES AND TRAJECTORIES OF SEPARATED STORES IN DISTURBED FLOW FIELDS

W. N. MacDermott and P. W. Johnson

ARO, Inc.

March 1973

Approved for public release; distribution unlimited.

PROPULSION WIND TUNNEL FACILITY
ARNOLD ENGINEERING DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
ARNOLD AIR FORCE STATION, TENNESSEE

NOTICES

When U. S. Government drawings specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, or in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Qualified users may obtain copies of this report from the Defense Documentation Center.

References to named commercial products in this report are not to be considered in any sense as an endorsement of the product by the United States Air Force or the Government.

AERODYNAMIC FORCES AND TRAJECTORIES OF SEPARATED STORES IN DISTURBED FLOW FIELDS

W. N. MacDermott and P. W. Johnson
ARO, Inc.

FOREWORD

The work reported herein was conducted by the Arnold Engineering Development Center (AEDC) under sponsorship of the Air Force Armament Laboratory (AFATL), Air Force Systems Command (AFSC), under Program Element 62602F. Technical monitor for AFATL was Major William Miller.

The results of this research were obtained by ARO, Inc. (a subsidiary of Sverdrup & Parcel and Associates, Inc.), contract operator of AEDC, AFSC, Arnold Air Force Station, Tennessee. The work was performed from August 16, 1971, through June 30, 1972, under ARO Project No. PW5280, and the manuscript was submitted for publication on September 21, 1972.

This technical report has been reviewed and is approved.

MAURICE A. CLERMONT
Major, CF
Research and
Development Division
Directorate of Technology

ROBERT O. DIETZ
Director of Technology

ABSTRACT

A vortex-lattice potential flow computer program capable of accepting nonuniform flow boundary conditions but previously restricted to incompressible flows with symmetry was modified to eliminate these restrictions. The program was structured in such a way that, after preliminary calculations of a purely geometric nature were performed one time for a given body, potential flow solutions for any set of boundary conditions on that body could be obtained in computer times measured in seconds rather than minutes. The aerodynamic characteristics of an M-117 bomb, represented by a network of 312 vortices, were calculated for uniform flow at a Mach number of 0.5 and were found to agree with wind tunnel measurements to within 10 percent, except for drag. The program was also used to compute forces on an M-117 bomb at a number of different locations in the disturbed flow field generated by an F-4C parent aircraft. In this case, absolute values of the force coefficients were generally in poor agreement with wind tunnel values, but the incremental variations of the calculated coefficients through the nonuniform flow field were within the range from 5 to 10 percent of wind tunnel measurements. A store separation routine was added to the potential flow program, and several representative store separation trajectories were calculated.

CONTENTS

Page

ABSTRACT iii
NOMENCLATURE viii
I. INTRODUCTION 1
II. DESCRIPTION OF ANALYTICAL METHODS
2.1 Potential Flow Calculations 2
2.2 Six-Degree-of-Freedom Store Trajectory Calculations 6
III. VORTEX-LATTICE AND STORE TRAJECTORY COMPUTER PROGRAM
3.1 Description of Program A 8
3.2 Description of Program B 10
3.3 Description of Program C 11
3.4 Computing Time 13
IV. RESULTS OF CALCULATIONS
4.1 Corrections to Calculated Force Coefficients 14
4.2 Correction of Experimental Force Data 16
4.3 Comparison of Calculations and Measurements in Uniform Flow 16
4.4 Comparison of Calculations and Measurements in F-4C Flow Field 17
4.5 Typical Store Separation Trajectories 19
V. SUMMARY 20
REFERENCES 23

APPENDIXES

I. ILLUSTRATIONS

Figure

1. Shape of M-117 Bomb
a. Dimensional Sketch of 1/20-Scale M-117 Bomb Model 27
b. 312-Vortex Model for Approximation of M-117 Bomb 27

Figure Page

2. Comparison of Calculated and Experimental Force Coefficients on M-117 Bomb, Uniform Flow
 a. C_N versus Angle of Pitch 28
 b. C_m versus Angle of Pitch 28

3. Effect of Mach Number on Calculated and Experimental Force Coefficients, Uniform Flow
 a. Calculated Normal-Force Coefficient 29
 b. Calculated Pitching-Moment Coefficient 29
 c. Calculated Axial-Force Coefficient 29

4. Locations of Bomb under F-4C Parent Aircraft at Which Calculated and Measured Forces Were Compared . . . 30

5. Calculated Variation of C_N and C_m with Pitch Angle in F-4C Flow Field, YP = -3.16, XP = 12.99, M = 0.50 31

6. Comparison of Absolute Values of Calculated and Experimental Force Coefficients in F-4C Flow Field, M = 0.5, YP = -3.16
 a. C_N versus ZP, Constant Pitch, Zero Yaw . . . 32
 b. C_m versus ZP, Constant Pitch, Zero Yaw. . . 32
 c. C_Y versus ZP, Constant Yaw, Zero Pitch. . . 33
 d. C_n versus ZP, Constant Yaw, Zero Pitch . . . 33

7. Comparison of Incremental Variation of Calculated and Experimental Force Coefficients in F-4C Flow Field, M = 0.5
 a. C_N versus ZP, Constant Pitch, Zero Yaw, YP = -3.16 34
 b. C_N versus ZP, Constant Pitch, Zero Yaw, YP = -4.66 34
 c. C_N versus ZP, Constant Pitch, Zero Yaw, YP = -6.16 35
 d. C_m versus ZP, Constant Pitch, Zero Yaw, YP = -3.16 35
 e. C_m versus ZP, Constant Pitch, Zero Yaw, YP = -4.66 36
 f. C_m versus ZP, Constant Pitch, Zero Yaw, YP = -6.16 36
 g. C_Y versus ZP, Constant Yaw, Zero Pitch, YP = -3.16 37

Figure | Page

7. Continued
   h. C_Y versus ZP, Constant Yaw, Zero Pitch, YP = -4.66 37
   i. C_Y versus ZP, Constant Yaw, Zero Pitch, YP = -6.16 38
   j. C_n versus ZP, Constant Yaw, Zero Pitch, YP = -3.16 38
   k. C_n versus ZP, Constant Yaw, Zero Pitch, YP = -4.66 39
   l. C_n versus ZP, Constant Yaw, Zero Pitch, YP = -6.16 39

8. Average Discrepancy Between Calculated and Experimental Force Coefficient, Incremental Basis, Pitch and Yaw = -4 deg and 6 deg, M = 0.5
   a. Average Discrepancy in C_N, C_m, C_Y, C_n, and Overall 40
   b. Variation of Average Theoretical/Experimental Discrepancy with ZP 40

9. Comparison of Incremental Variation of Calculated and Experimental Force Coefficients in F-4C Flow Field, M = 0.85
   a. Best Comparison, C_Y versus ZP, Constant Yaw, Zero Pitch, YP = -6.16 . . . 41
   b. Worst Comparison, C_m versus ZP, Constant Pitch, Zero Yaw, YP = -6.16 . . . 41

10. Average Discrepancy Between Calculated and Experimental Force Coefficients, Incremental Basis, Pitch and Yaw = -4 deg and 6 deg, M = 0.85
   a. Average Discrepancy in C_N, C_m, C_Y, C_n, and Overall 42
   b. Variation of Average Discrepancy with ZP . . 42

11. Comparison of Separation Trajectories with Uniform Flow Force Coefficients and Nonuniform Flow Force Coefficients, F-4C Aircraft, 250-lbm M-117 Bomb, on No. 2 TER Station, Inboard Pylon, M = 0.50, 5000-ft Altitude, Pitch Angle of A/C = 0.30 deg, 1000-lb Ejector Force at 45 deg to Vertical 43

II. USER'S GUIDE TO COMPUTER PROGRAM 44

NOMENCLATURE

$A^{(1)}$, $A^{(2)}$	Submatrices of the H^{-1} matrix, defined in Section 3.2
$\vec{B}$	Matrix of direction cosines of unit normals to surface
ΔC_A	Base pressure correction, Eq. (20)
$\Delta C_{A_{base}}$	Correction to calculated axial-force coefficient for a base force, Eq. (16)
$\Delta C_{A_{fric}}$	Frictional correction to axial-force coefficient, Eq. (19)
C_A, C_N, C_Y	Axial, normal, side-force coefficients, respectively
C_F	General force coefficient
C_ℓ, C_m, C_n	Rolling, pitching, yawing-moment coefficients, respectively
C_p	Pressure coefficient
$\vec{F}_i$	Force on ith segment of a horseshoe vortex, defined by Eq. (10)
$\vec{G}^{(1)}$, $\vec{G}^{(2)}$	Submatrices of the $\vec{G}$ matrix, defined in Section 3.1
$\vec{G}_{k,j}$	Geometric influence coefficient, Eq. (14), Ref. 1.
H	Matrix of geometric factors defined by Eq. (8)
$H^{(1)}$, $H^{(2)}$	Submatrices of the H matrix, defined in Section 3.2
M	Mach number
N_j	Number of horseshoe vortices in a network representing a surface
NUFF	Nonuniform flow field
$\vec{n}$	Unit vector normal to surface
p	Static pressure
$\vec{q}_{k,j}$	Velocity at field point k induced by vortex j, Eq. (5)
Re_x	Reynolds number based on model length
S_{ref}	Reference area, 0.5025 in.^{2}
$\vec{U}_\infty$	Velocity in flow field undisturbed by store
$\vec{V}$	Velocity in flow field, including induced velocity, $= \vec{U}_\infty + \Sigma\vec{q}$, Eq. (15)

X, Y, Z	Cartesian coordinates in parent aircraft reference system, positive forward, to the right, and down, as seen by pilot
XP, YP, ZP	Coordinates of nose of store model in parent aircraft coordinate system
x, y, z	Cartesian coordinates in store reference system
β	$\sqrt{1 - M_\infty^2}$
Γ_j	Strength of horseshoe vortex j
ν	Angle of pitch, positive—nose up
ρ	Density
Φ	Velocity potential
Ψ	Angle of yaw, positive—nose right

SUBSCRIPTS

b	Base
base	Base of bomb
comp	Compressible flow
cross	Cross-sectional area
i	Straight-line segment of a horseshoe vortex
inc	Incompressible flow
j	Horseshoe vortex
k	Field point at which velocity is calculated
plan	Planform area
S	Area
∞	Free-stream condition

SECTION I
INTRODUCTION

In recent years, testing of aircraft store separation characteristics has emerged as one of the major uses of developmental wind tunnels. As a consequence of the many possible configurations and flight conditions of interest, such testing is characterized by a voluminous production of test data. To assist in the evaluation of tunnel data and in its extrapolation to full scale, several analytical techniques of store separation trajectory calculation have been under development. These techniques possess varying degrees of dependence on experiment, from nil to a major dependence. Because of shortcomings of aerodynamic theory and present digital computer limitations, any completely theoretical calculation apparently will remain an unattainable ideal for a number of years in the future. An analytical method which places a major reliance on wind tunnel measurements is the so-called "grid method," in which force characteristics on a given store are measured at a large number of points within a volume beneath a given parent aircraft, stored in some accessible form, and finally are used as a source of force coefficient data (interpolated) at points along a distinct trajectory developed by double integration of the equations of motion. This technique has achieved a modicum of use, but is limited to a single store/aircraft combination for a given set of experimental measurements.

During the past two years, the staff of the AEDC Propulsion Wind Tunnel (PWT) has attempted to develop an alternate technique having one step less dependence on wind tunnel measurements. This technique is built around the use of potential flow solutions for a given store, using as boundary conditions on the store the disturbance flow angles measured in the nonuniform flow field for a given parent aircraft. Ideally, this approach would allow a single set of wind tunnel measurements for a single parent aircraft to be used with any number of different store calculations.

As reported in Ref. 1, a potential flow program based on vortex singularities was developed in a form allowing imposition of boundary conditions based on known nonuniform flow fields. Considerable effort was expended in developing a vortex network representation of the M-117 bomb which would produce calculated force characteristics sufficiently close to values measured in the wind tunnel. A network composed of 140 vortices representing one half of the M-117 bomb shape and 16 vortices representing a wake at the base of the bomb (Fig. 1, Appendix I) was found to give 10-percent agreement (except for drag) with force measurements made in a uniform flow. This model was then used to calculate force characteristics on the M-117 bomb at various locations within the nonuniform flow field (NUFF) of the F-4C aircraft. This calculation was made only in an approximate sense, since only the downwash components of the NUFF were allowed for, and even these were averaged laterally. In this approximation, the absolute values of the calculated force coefficients differed appreciably from wind tunnel measurements, but the incremental behavior of the calculated force coefficients within the NUFF showed the same trends as did the wind tunnel measurements. Based on this observation, the project effort was continued, and the results of subsequent calculations are presented in this report.

SECTION II
DESCRIPTION OF ANALYTICAL METHODS

2.1 POTENTIAL FLOW CALCULATIONS

2.1.1 Boundary-Value Problem for a Velocity Potential

As discussed in Ref. 2, the analytical description of the flow field surrounding an aerodynamic body can be obtained from a solution of the differential equation for a velocity potential, i.e., Laplace's equation, in the incompressible case,

$$\nabla^2\Phi = 0 \tag{1}$$

This assumes steady, irrotational, inviscid, attached flow. In the general case of compressible flow, the differential equation of the velocity potential is highly nonlinear, and exact solutions are not generally possible to obtain. Almost all work in the compressible regime has been done in the linearized approximation assuming perturbations to the free stream are small. The velocity potential, Φ, then satisfies

$$\left(1 - M_\infty^2\right)\frac{\partial^2}{\partial X^2}\Phi + \frac{\partial^2}{\partial Y^2}\Phi + \frac{\partial^2}{\partial Z^2}\Phi = 0 \tag{2}$$

the Neumann condition on the planform surface

$$\vec{V}\cdot\vec{n} = -\nabla\Phi\cdot\vec{n} = 0 \tag{3}$$

and the boundary condition at infinity

$$\nabla\Phi = -\vec{U}_\infty \text{ at } X = Y = Z = \pm\infty \tag{4}$$

Since the Laplace equation is linear, superposition of elementary solutions (sources, sinks, doublets, and vortices) will allow buildup of an incompressible flow over quite general shapes satisfying the boundary conditions, Eqs. (3) and (4).

Since it was desired to allow for trailing vorticity in the flow, caused by three-dimensional lifting surfaces, a potential flow program was developed based on superposition of vortex elements alone, excluding use of source, sinks, and doublets. This program was modelled after that of Ref. 3.

Prandtl introduced the concept that a solid body immersed in a flow is equivalent in external flow to a sheet of continuously distributed vorticity at the surface of the body. A relation exists between the velocity at any point and the strength and orientation of the distributed vorticity. Since this relationship cannot be solved in closed form, it is necessary, for computational reasons, to replace the distributed vortex sheet representation of a solid body (an exact concept) with a series of concentrated vortex elements having the same net strength (an approximation), for which a simple relationship does exist. This relationship is identical with the Biot-Savart law, which describes the magnetic field induced by an electric current flowing through a conductor. In its fluid-mechanical formulation, the Biot-Savart law is a differential relationship (see Eq. (7) of Ref. 1)) which can be integrated in closed form along a straight-line segment of a vortex (the description of a vortex segment is given in Fig. 4c of Ref. 1). The result for a single horseshoe vortex composed of a number of straight-line vortex segments can be expressed as

$$\vec{q}_{k,j} = \vec{G}_{k,j}\Gamma_j \tag{5}$$

where $\vec{q}_{k,j}$ is the velocity at field point k due to vortex j, Γ_j is the strength of vortex j, and $\vec{G}_{k,j}$ is a geometric influence coefficient representing a summation over all segments of the horseshoe vortex given in functional form by Eq. (14) of Ref. 1.

The strengths of the individual vortices, Γ_j, are determined by constraining the flow to be directed in a prescribed manner. Generally this is done by requiring the flow to be tangent to the surface at a set of points referred to as boundary points. The criterion for boundary point location used in obtaining the results described in this report combines simplicity and plausibility - the boundary point coordinates are taken to be centroids of quadrilateral panels formed by overlapping horseshoe vortices. A single boundary condition is expressed as

$$\vec{V}_k \cdot \vec{n}_k = \left(\vec{U}_{\infty_k} + \sum_{j=1}^{N_j} \vec{G}_{k,j} \Gamma_j \right) \cdot \vec{n}_k = 0 \tag{6}$$

which can be recognized as Eq. (3) written at a specific boundary point. The system of N_j equations representing all the required boundary conditions is thus

$$\sum_{j=1}^{N_j} \left(\vec{n}_k \cdot \vec{G}_{k,j} \right) \Gamma_j = - \vec{n}_k \cdot \vec{U}_{\infty_k} \qquad k = 1, \ldots, N_j \tag{7}$$

It is convenient to introduce the matrix notation

$$\left[H_{k,j} \right] \equiv \left[\vec{n}_k \cdot \vec{G}_{k,j} \right] \tag{8}$$

giving

$$\left[H_{k,j} \right] \left\{ \Gamma_j \right\} = \left\{ -\vec{n}_k \cdot \vec{U}_{\infty_k} \right\} \tag{9}$$

as the matrix equation to be solved for the unknown vortex strengths. Upon solving Eq. (9) by inversion of the H matrix, the values of vorticity, Γ_j, are substituted into Eq. (5) to obtain the velocity distribution at the field points.

The distribution of aerodynamic forces on a planform is obtained by application of the Kutta-Joukowski law

$$\vec{F}_i = \rho \vec{V}_i \times \vec{\Gamma}_i \tag{10}$$

on each segment i of every vortex j. Finally, summation over all vortex segments is performed to obtain the forces that are used in the definitions of aerodynamic force coefficients. A precaution must be observed in the application of Eq. (10); namely, the velocity vector $\vec{V}_i$ should not include any induced effect of the ith vortex element. That is, $\vec{V}_i$ should be the flow field velocity vector which would exist in the absence of the vortex element, but at the physical location of the element usually assumed to be its midpoint.

The foregoing method of construction of a potential flow is valid whatever the nature of $\vec{U}_{\infty_k}$ in the boundary conditions shown in Eq. (7). In cases of uniform onset flow, it is a constant vector. In the case of present interest, disturbed flow fields, it is assumed to vary over the surface of the body and is obtained by experimental flow-field measurements in the absence of the store. This approach is legitimate only in the case of simple interference; that is, a large body (parent aircraft) produces a disturbance flow felt by a smaller body (store), but not vice versa. In case the disturbance field of the small body is also felt at the larger body, mutual interference is said to occur, and solution of the potential flow problem is required simultaneously for the two bodies. In the simple interference case it is necessary to solve the potential flow problem for only the smaller body. In practice this means that the present approach is strictly valid only beyond a certain distance of separation defining the limits of mutual interference.

2.1.2 Compressibility Corrections to Incompressible Flow Calculations

Since the Biot-Savart law is valid only for incompressible flow, the vortex-lattice calculations are performed as solutions of the exact equation for the velocity potential of an incompressible flow, Eq. (1). Allowance for compressibility is then made by use of the Goethert similarity rule (Ref. 2), which relates a compressible flow (linearized approximation, Eq. (2)) to a corresponding incompressible flow about an affinely related shape which is decreased in all lateral dimensions by the factor $\beta = \sqrt{1 - M_\infty{}^2}$. The rule states that the pressure coefficients at corresponding points in these two flows are related by

$$C_{p_{comp}} = \frac{1}{\beta^2} C_{p_{inc}} \tag{11}$$

and, for characteristic force coefficients obtained by use of a reference area which is equal to or proportional to the area over which ΔC_p is integrated to obtain the force (or moment),

$$C_{F_{comp}} = \frac{1}{\beta^2} C_{F_{inc}} \tag{12}$$

For geometrically similar bodies, all areas related to the body are proportional to all other areas, and there are no qualifications required by the choice of reference area. The Goethert rule is not applied to geometrically similar bodies, however, and the form of the force coefficient correction, Eq. (12), does depend on the specific reference area chosen. For slender, axisymmetric shapes such as the M-117 bomb, the cross-sectional area is usually chosen as the reference area, and this area is proportional to the area of integration of C_p only for the axial-force coefficient. All other force coefficients are formed by integration over a planform area; thus,

$$C_{F_{comp}}\left(\frac{S_{plan}}{S_{cross}}\right)_{comp} = \frac{1}{\beta^2}\left(\frac{S_{plan}}{S_{cross}}\right)_{comp}\left(\frac{S_{cross}}{S_{plan}}\right)_{inc}\left(\frac{S_{plan}}{S_{cross}}C_F\right)_{inc}$$

$$\left(C_{F_{comp}}\right)S_{cross} = \frac{1}{\beta^2}\,\frac{\left(S_{plan}\right)_{comp}}{\left(S_{plan}\right)_{inc}} \times \frac{\left(S_{cross}\right)_{inc}}{\left(S_{cross}\right)_{comp}}\left(C_{F_{inc}}\right)S_{cross}$$

$$\left(C_{F_{comp}}\right)S_{cross} = \frac{1}{\beta^2}\left(\frac{1}{\beta} \times \beta^2\right)\left(C_{F_{inc}}\right)S_{cross}$$

$$\left(C_{F_{comp}}\right)S_{cross} = \frac{1}{\beta}\left(C_{F_{inc}}\right)S_{cross} \qquad (13)$$

Equation (13) was used for correction of normal-force, side-force, pitching-moment, yawing-moment, and rolling-moment coefficients. It is noted that, in application to NUFF flows, the NUFF is also contracted by the factor β in all directions transverse to the flow.

2.2 SIX-DEGREE-OF-FREEDOM STORE TRAJECTORY CALCULATIONS

A routine was added to the potential flow computer program to allow trajectory calculations through nonuniform flow fields. The trajectory computer program described in this report incorporates the same differential equations in describing six-degree-of-freedom motion as those used in the AEDC/PWT Captive Trajectory System (CTS). The differences between the two approaches are primarily that (1) each program uses different numerical integration techniques and (2) the program of this report uses theoretical force coefficients, whereas the CTS program uses experimentally determined forces.

2.2.1 Equations of Motion

The differential equations describing conservation of linear momentum are Eqs. (II-34), (II-35), and (II-36) of Ref. 4; the angular momentum relationships are given by Eqs. (II-37), (II-38), and (II-39) of Ref. 4. (Equation (II-37) contains a misprint; see Eq. (11) of Ref. 5.) The store rotational sequence is assumed to be pitch-yaw-roll. The equations resulting from this assumption relating store linear velocity in wind axes coordinates to velocity in body axes coordinates are Eqs. (II-13), (II-14), and (II-15) of Ref. 4. (These equations also contain a misprint; the transformation matrix, A, should be replaced by the transpose of A.) The differential equations relating pitch, yaw, and roll velocities to store rotational velocities about body axes are Eqs. (II-4), (II-6), and (II-8) of Ref. 4. See also page 23, case 4, of Ref. 5 for the coordinate transformations used in the above expressions.

2.2.2 Numerical Integration Method

The motivation that led to the adoption of the particular numerical integration scheme incorporated in the trajectory program of this report was optimization of computer execution time consistent with the accuracy of its results. It was adjudged that accurate results could be obtained using the largest possible step size (in time) from the classical fourth-order Runge-Kutta method. To apply the Runge-Kutta algorithm in its usual form would require transferal of data (the H^{-1} and $\vec{G}$ matrices) from external memory into the computer core each time a single set of force coefficients is to be determined. (Since the size of the H^{-1} and $\vec{G}$ matrices is such that they cannot be stored in-core in their entirety, they must be input repeatedly.) As the data input operation constitutes a significant fraction of the total time required to perform vortex-lattice calculations (and, consequently, trajectory calculations), a procedure has been adopted to compute a number of sets of force coefficients each time the H^{-1} and $\vec{G}$ matrices are input. In order to accomplish this saving in execution time it was necessary to devise a scheme that would incorporate this approach in a modified Runge-Kutta algorithm. The procedure that was constructed is a predictor-corrector method and is described in the following.

A first approximation to the solution for a trajectory is obtained by performing Runge-Kutta calculations over a specified number of time steps (two in the solution subsequently reported herein) using extrapolated values of force coefficients. The values of store orientation obtained in this manner are used to recompute the force coefficients, this time by the vortex-lattice method. The Runge-Kutta calculations are then performed a second time. The force coefficients used in this second approximation to a trajectory are obtained in the form of Taylor series expansions, where the leading terms in the expansions are the vortex-lattice results. The correction term to the drag force is assumed to be a quadratic function of the store angle of incidence with respect to the free stream; the correction terms to the normal force, side force, pitching moment, and yawing moment are all assumed to be proportional to computed values of the static stability derivatives.

SECTION III
VORTEX-LATTICE AND STORE TRAJECTORY COMPUTER PROGRAM

The determination of aerodynamic forces by the vortex-lattice method has been performed by a digital computer. The overall set of calculations has been divided into three separate programs; for descriptive purposes, these will be referenced in this report as Programs A, B, and C. The reasons for this subdivision are twofold:

1. For a given vortex network and Mach number, certain of the calculations are of a purely geometric nature and consequently are performed only one time for a given store (in Programs A and B) and then are stored on magnetic tape, to be used in conjunction with parametric variations of store angle of incidence and location in a surrounding NUFF (in Program C);
2. The computer requirements for performing the geometric calculations, in terms of internal memory and execution time, are of such a magnitude that these computations cannot all be performed in one program; hence, they are performed in two separate programs (A and B).

Program C, which computes aerodynamic characteristics (the distributions of velocity, pressure coefficient, and force coefficients, as well as the total-force coefficients on a store), is written in two separate versions. One of these computes the variables just mentioned for a specified set of spatial orientations of a store with respect to its parent aircraft where the store is assumed to be motionless with respect to wind axes; the other version solves for both the force coefficients and the trajectory of a store separated from its parent aircraft. The portion of the computer code that pertains to the trajectory of a separated store is based upon the nomenclature and equations used in Ref. 4.

A brief description and user's guide of these programs follows. A more detailed description may be obtained from the authors.

3.1 DESCRIPTION OF PROGRAM A

This program consists of a MAIN program and SUBROUTINE ICOEFF. The principal results are the calculation of the $\vec{G}$ and H variables and the transferal of these data to magnetic tape.

The $\vec{G}$ matrices are geometric influence coefficients, the resultant values of which are equal to the sum of contributions from the bound spanwise vortex segments (denoted by the FORTRAN name DIC), bound chordwise segments (EIC), and trailing vortices (FIC). The H matrix is computed from

$$H = \vec{B} \cdot \vec{G}_1 \tag{14}$$

where $\vec{B}$ is the matrix of direction cosines of the unit normals to the surface at the boundary points. $\vec{G}_1$ is the value of $\vec{G}$ obtained when the influence of vortices at boundary points is considered. (It may be noted that Eq. (14) is equivalent to Eq. (8), the only difference being the use of alternate notations.)

The first step leading to the calculation of influence coefficients is the specification of a network of discrete line vortices on the surface of the aerodynamic planform being analyzed. A general description of vortex networks is presented in Section III of Ref. 3 and Fig. 4 of Ref. 1. The specific application to the M-117 bomb that was used to obtain the results of this report is described in Sections 4.3 and 5.1.1 of Ref. 1 and is presented in Fig. 1 of this report.

The equations for G are shown combined in succinct notation in Eq. (14) of Ref. 1; they are presented in the detailed form in which they are incorporated in the computer code in Eqs. (2a), (4a), (7a), (10a), and (12a) of the Appendix to Ref. 3.

In general, all matrices of geometric factors are computed in partitioned form if the planform being analyzed possesses geometric symmetry about its longitudinal axes. In the special case of zero yaw attitude combined with lateral symmetry of the flow field about the store, the matrices can be further simplified (see p. 18 of Ref. 1). In particular, it can be shown in the general case that the x, y, and z components of $\vec{G}$ are

$$\left[\begin{array}{c|c} G_x^{(1)} & G_x^{(2)} \\ \hline -G_x^{(2)} & -G_x^{(1)} \end{array}\right], \quad \left[\begin{array}{c|c} G_y^{(1)} & G_y^{(2)} \\ \hline G_y^{(2)} & G_y^{(1)} \end{array}\right], \quad \left[\begin{array}{c|c} G_z^{(1)} & G_z^{(2)} \\ \hline -G_z^{(2)} & -G_z^{(1)} \end{array}\right]$$

where $\vec{G}^{(1)}$ contains the influence of vortices on the +y (or -y) side of the x-z symmetry plane at field points on the +y (or -y) side; $\vec{G}^{(2)}$ contains the influence of vortices on the +y (or -y) side at field points on the -y (or +y) side. The partitioned form of H is discussed in the following section.

3.2 DESCRIPTION OF PROGRAM B

The purpose of Program B is twofold: (1) to perform the inversion of the H matrix, and (2) to create a magnetic tape on which the matrices HDIAG, H^{-1}, H, and $\vec{G}$ are written. (HDIAG is a column matrix whose elements are formed from the principal diagonal of H.)

Prior to inversion, the elements of the H matrix are normalized by the elements of its principal diagonal. Thus the resulting matrix has the value of unity everywhere on the principal diagonal and is written in partitioned form (see Eq. (26) of Ref. 1):

$$H = \left[\begin{array}{c|c} H^{(1)} & H^{(2)} \\ \hline H^{(2)} & H^{(1)} \end{array}\right]$$

Likewise, it can be shown that the inverse of H is written in partitioned form (see Eq. (28) of Ref. 1):

$$H^{-1} = \left[\begin{array}{c|c} A^{(1)} & A^{(2)} \\ \hline A^{(2)} & A^{(1)} \end{array}\right]$$

The FORTRAN code written to perform the calculation of $A^{(1)}$ and $A^{(2)}$ has been devised with the objectives of (1) being able to invert the largest possible matrix within the limitations imposed by the size of the available core, and (2) minimizing the computer time required to execute the calculations. The salient feature of the algorithm that has been derived to satisfy these mutual requirements is that storage locations are reserved in-core to accommodate only one square matrix. The size of this matrix is NVORT rows x NVORT columns, where NVORT is the number of vortices in the network on one side of the x-z plane of symmetry of a planform. In addition, an option has been provided for segmenting the program into two separately executed submittals. This is to allow for those instances in which the size of the H matrix is so large that an excessive amount of computer time would be required to perform the inversion from beginning to end in one submittal.

In order to facilitate the reading of the computer code for Program B, a summary of the sequence of the intermediate calculations is presented for the special case where the entire inversion is performed in one submittal of the program:

```
Normalize $H^{(1)}$ by the elements of its principal diagonal (HDIAG)
WRITE(21) HDIAG
Compute $H^{(1)^{-1}}$
Normalize $H^{(2)}$ by the elements of the principal diagonal of $H^{(1)}$
Compute $H^{(2)}H^{(1)^{-1}}$
Compute $H^{(2)}H^{(1)^{-1}}H^{(2)}$
Compute $H^{(1)} - H^{(2)}H^{(1)^{-1}}H^{(2)}$
Compute $A^{(1)} \equiv \left[H^{(1)} - H^{(2)}H^{(1)^{-1}}H^{(2)}\right]^{-1}$ and WRITE(21) $A^{(1)}$
Compute $A^{(2)} \equiv -A^{(1)}H^{(2)}H^{(1)^{-1}}$ and WRITE(21) $A^{(2)}$
WRITE(21) $H^{(1)}, H^{(2)}, \vec{G}$
```

3.3 DESCRIPTION OF PROGRAM C

This program uses the output from Program B to compute aerodynamic characteristics and (as an option) separated store trajectories. The computer requirements in terms of internal memory are of such magnitude that it is necessary to subdivide the program into three separately compiled job steps. Each of these will be described in turn.

3.3.1 First Job Step

The purpose of this step is to transfer a subset of the data written on magnetic tape in Program B onto a direct-access device. The purpose of this operation is to achieve faster execution times in the third job step.

There are two options which may be exercised in this step. If it is desired to show by substitution (a posteriori in the third job step) that the computed values of vortex strengths satisfy the matrix equation for $\{\Gamma_j\}$, Eq. (9), then the H matrix is transferred from tape to disk; otherwise, it is not. The second option concerns the calculation of the pressure coefficient distribution (in the third step). If it is desired to perform this calculation, the influence coefficients at boundary points are transferred from tape to disk; otherwise, they are not.

3.3.2 Second Job Step

The primary functions of this step are as follows: (1) to read input data, which comprise values for parameters of the problem and the NUFF, from punched cards and to pass this data to the following job step, (2) to determine the coordinates of the line segments comprising the vortex network, and (3) to compute the locations of the control points at which the flow-field vector is constrained to be in a prescribed direction. (Items 2 and 3 are also computed in Program A.)

3.3.3 Third Job Step

The FORTRAN code for this step is composed of a MAIN program and subroutines VORLAT, FRESTR, VELOCY, ACOEFF, and AXES. There are two optional versions of this MAIN program, depending upon whether or not it is desired to perform trajectory calculations. The subroutines are all identical for these two versions.

The primary purpose of the trajectory version of the MAIN program is to provide values for many of the parameters which enter into the equations of motion, and then to numerically integrate these equations. The other version has been coded for the purpose of defining values of variables used in calculations in the subroutines.

Vorticity and velocity distributions are computed in SUBROUTINE VORLAT. Values of Γ_j are given by the solution of Eq. (9) by matrix inversion, and the formula for velocity at the field point k is (see Eq. (15) of Ref. 1)

$$\vec{V}_k = \vec{U}_{\infty_k} + \sum_{j=1}^{N_j} \vec{G}_{k,j}\Gamma_j \qquad (15)$$

The purpose of SUBROUTINE FRESTR is to interpolate on experimental values of the NUFF to determine the downwash, sidewash, and magnitude of velocity at points on the surface of the store. Three sets of such interpolations are performed; the first set of interpolated values is obtained at the boundary points, the second at the midpoints of spanwise segments of the vortex network, and the third set at midpoints of chordwise segments.

No calculations are performed in SUBROUTINE VELOCY. The sole purpose of this subroutine is to print (as an option) the distribution of velocities.

SUBROUTINE ACOEFF performs aerodynamic calculations. As an optional capability the pressure coefficient can be computed at each of the boundary points, using Eqs. (23) and (24) of Ref. 1 (see also Section IV-h of the Appendix to Ref. 3). The distribution of aerodynamic forces on a planform is determined using the Kutta-Joukowski law, Eq. (10), and these forces are then summed. The detailed equations to accomplish these results, which are incorporated in the computer code, are given in Eqs. (IV-a) through (IV-g) of Ref. 3.

The coordinate transformations relating components of variables in wind axes to the components in a body axis reference system utilize the coefficients computed in SUBROUTINE AXES. The formulas for these coefficients can be found on p. 31 of Ref. 4.

3.4 COMPUTING TIME

The vortex-lattice solutions reported herein were obtained on an IBM 370/155 digital computer whose Central Processing Unit (CPU) can store a source program of 45,000 decimal words using internal memory. Examples will be cited of computational times required to execute solutions. The time required varies with the number of vortices, since the execution time is a function of the size of the matrices containing the geometric factors and the number of elements in these matrices varies approximately as the number of vortices squared. A representative calculation assuming 156 vortices on each side of the plane of geometric symmetry and asymmetry of flow about the x-z plane, equivalent to a 312-vortex problem, requires 10 min to execute Program A, and 14 min to execute Program B.

The time required to execute Program C is virtually all consumed in performing the calculation of the force coefficients in the third job step of this program. Since a significant amount of this time is expended in transferring data from external memory into the core of the CPU, a procedure to optimize the calculations has been devised which consists of computing the force coefficients for a number of cases each time the data are read from external memory. (A case means one orientation of a store with respect to the parent aircraft; the external memory used is a direct access device.) The maximum number of cases that can be computed each time the external memory is read is limited by the amount of core available to store intermediate results. In the present application the programmed procedure is to read the direct access device, compute five cases (the number of cases is an input to Program C), rewind the direct access device, and repeat this sequence any desired number of times. It is found that the execution of five cases requires 1 min 25 sec; therefore, each potential flow solution requires an average of 17 sec of CPU time.

The particular option of the numerical method devised to compute trajectories that has been exercised to produce the results of Section 4.5 requires five vortex-lattice cases to be computed for each advance of two time steps in the numerical integration of the equations of motion. Thus, for example, in the sample trajectories reported herein in which each of 16 time steps was specified to be 0.05 sec of real time, a trajectory of 0.8 sec total real time required 11 min of CPU time.

SECTION IV
RESULTS OF CALCULATIONS

As shown in Fig. 1, the physical shape of the M-117 bomb body was represented by a system of horseshoe vortices having the "trailing" elements running forward to the nose tip, whence they became superimposed and trailed back downstream to infinity along the bomb axis. The fins were modeled by horseshoe vortices having trailing segments in the usual direction. To obtain a reasonable degree of agreement with experiment, it was found necessary to impose a wakelike character on the flow in the base region by extension of the vortex network downstream of the actual bomb body (Ref. 1). These wake vortices were used only to control the velocity pattern near the base of the bomb; forces on these vortices were not included as force on the bomb.

4.1 CORRECTIONS TO CALCULATED FORCE COEFFICIENTS

To ensure that the calculated forces represented the best approach to reality, three different corrections were added to the force coefficients resulting from the potential flow calculations.

4.1.1 Base Force Correction

As described in Ref. 1, summation of the Kutta-Joukowski forces over the vortex network (with wake vortices excluded from the summation) results in a large, unbalanced internal pressure force and a net thrust on the network. This occurs because the vorticity representing the solid base and, hence, the internal and external force on the base, is not accounted for. To allow for this neglected base force, a correction to the calculated axial-force coefficient is given by

$$\Delta C_{A_{base}} = \frac{S_{base}}{S_{ref}} \left[C_{p_i} - C_{p_b} \right] \tag{16}$$

For the purpose of comparison with wind tunnel data, the usual reduction to zero base drag conditions, $C_{p_b} = 0$, is made. The pressure coefficients inside the vortex network in the base region were observed to vary with angle of pitch or yaw and were given approximately by the curve-fit:

$$C_{p_i} = 0.992 - 0.00018 \ \text{x} \ (\nu \text{ or } \Psi, \text{deg})^2 \tag{17}$$

The base-force correction can be seen to be very nearly the stagnation pressure acting over the base area. Lastly, the base area in this case was taken to be the area of the octagonal base of the vortex network, rather than the area of the circular base of the actual bomb shape. The resulting correction was thus given by

$$\Delta C_{A_{base}} = (0.495 \ \text{x} \ 0.9)(0.992 - 0.00018) \ \text{x} \ (\nu \text{ or } \Psi, \text{deg})^2 \tag{18}$$

4.1.2 Compressibility Correction

After the addition of the base-force correction, all force coefficients calculated for incompressible flow over the (possibly contracted) body shape were corrected for compressibility by Eq. (12) or (13).

4.1.3 Skin-Friction Correction

Finally, an approximate allowance was made for contribution of skin friction to axial force:

$$\begin{aligned} \Delta C_{A_{fric}} &= 0.0316 \text{ for a laminar boundary layer} \\ &= 0.0935 \text{ for a turbulent boundary layer} \end{aligned} \tag{19}$$

These corrections were based on flat plate skin friction at $Re_x = 1.1 \times 10^6$, corresponding to the wind tunnel Reynolds number at M = 0.5. Since drag exerts only a minor influence on separation trajectories, high accuracy on C_A was not considered important, and the skin-friction corrections were simply applied as constants at all conditions.

4.2 CORRECTION OF EXPERIMENTAL FORCE DATA

Because of the marked sensitivity of base-region flows to Reynolds number, experimental axial-force data on blunt-base bodies are usually corrected to a standard condition of zero base drag, defined as the condition $p_b = p_\infty$, or $C_{p_b} = 0$. The standard correction expression for this condition is

$$\Delta C_A = C_{p_b} \frac{S_{base}}{S_{ref}} \tag{20}$$

Since base pressure measurements were not made during the test reported in Ref. 6, it was simply assumed that the base pressure was equal to pressure on the bomb just forward of the base, as given by the vortex-lattice calculations,

$$\Delta C_A = -0.065 \times 0.495 = -0.0322$$

4.3 COMPARISON OF CALCULATIONS AND MEASUREMENTS IN UNIFORM FLOW

Calculated force coefficients and measured force coefficients on the M-117 bomb in uniform flow are compared in Figs. 2 and 3. Both theory and experiment have been adjusted to as nearly the same basis of comparison as possible by use of the corrections discussed above.

In Fig. 2, the normal-force and pitching-moment coefficients are plotted versus angle of pitch. Theoretical curves are given for M = 0, 0.5, and 0.85. Experimental data are given for M = 0.5 and 0.85. A general agreement of about 10 percent exists at M = 0.5.

The effect of Mach number on the theoretical-experimental force coefficient comparison is given in Fig. 3. The calculated normal-force coefficient at ν = 10 deg (Fig. 3a) displays a nearly constant 10-percent displacement from the measured values. The calculated pitching-moment coefficient at ν = 10 deg (Fig. 3b) however, displays an appreciably greater influence of Mach number than does the experimental data. The calculated axial-force coefficient with laminar boundary layer at zero pitch (Fig. 3c) is 17 percent greater than the wind tunnel measurement at M = 0.5, but the theoretical drag rise given by Eq. (12) is much greater than observed in the wind tunnel data. At M = 0.85, the calculated axial-force coefficient, corrected for compressibility, is 3.5 times the measured value.

4.4 COMPARISON OF CALCULATIONS AND MEASUREMENTS IN F-4C FLOW FIELD

The force characteristics of the M-117 bomb were calculated for twelve different locations of the bomb in the disturbed flow field of the F-4C parent aircraft, three spanwise locations and four vertical locations (Fig. 4). The outboard pylon is shown in Fig. 4 in a vertical orientation, which corresponds to that of the full-scale aircraft when carrying an external fuel tank. The flow-field data of Ref. 6, however, on which the present results are based, were obtained for a model having the outboard pylon canted outboard at 7.5 degrees. The flow-field data used as boundary conditions were obtained from the test reported in Ref. 6. These calculations were similar to those reported in Ref. 1, with the following exceptions:

1. All three components of the nonuniform flow field, not just downwash angles, were used as boundary conditions on the bomb.
2. Both yaw and pitch orientations (but not in combination) were included.
3. Compressibility corrections to the potential flow calculations were included.
4. Bomb model was rolled to put fins in vertical and horizontal planes, corresponding to the configuration used in the wind tunnel.

4.4.1 Comparison at M = 0.50

The variation of normal-force and pitching-moment coefficients with pitch angle for ZP = 3.13 to 9.13 in. at YP = -3.16 in. and M = 0.50 is given in Fig. 5. The uniform flow force characteristics are also presented for comparison. In Fig. 6 are given C_N, C_m, C_Y, and C_n versus the vertical separation distance, ZP, at constant pitch angles, zero yaw, and YP = -3.16 in. The absolute values of the calculated coefficients are not in good agreement with the measured values, although the calculated force coefficients are in slightly better agreement than are the moment coefficients. The discrepancies are on the order of 40 to 50 percent, and the compressibility corrections do not result in any systematic reduction of the differences.

If, however, the incremental effects of the NUFF on the force coefficients are considered, there is clearly a better agreement between the calculations and the measurements. In almost every case in Fig. 6, the shapes of the theoretical curves are quite similar to the shapes of the experimental curves. This fact is demonstrated for all the cases calculated in Figs. 7a through 1, in which the theoretical results have been arbitrarily adjusted to the experimental values at the maximum distance of separation from the parent aircraft, ZP = 9. 13 in. These data cover all twelve locations of the bomb and the four force coefficients, C_N, C_m, C_Y, and C_n. Throughout all of these plots the trend of the individual coefficient with ZP is well predicted by the potential flow calculations. The largest single discrepancy is 26 percent, in C_m at YP = -3. 16, ZP = 3. 13, and ν = 2 deg, but the average discrepancy is from 4 to 6 percent (Fig. 8). Variation of average discrepancy with vertical separation distance is also shown in Fig. 8. The increasing discrepancy at small separations is probably the result of mutual interference, which is not accounted for in the potential flow calculations. By extrapolation, it is estimated that the average discrepancy does not exceed 10 percent until the separation distance is about one-half the bomb diameter (0. 4 in.). The compressibility correction appears to be slightly beneficial at large separation, but it becomes of negative value in the mutual interference region.

This result, that incremental behavior of force coefficients is well predicted by calculations, suggests the possibility of a combined theoretical/experimental approach in which experimental force coefficients measured at some convenient location away from a parent aircraft are adjusted by means of potential flow results. Ideally, the experimental force coefficients would be measured only in the free stream. In the present case, values measured at ZP = 9. 13 in. are used as a reference because of an unexplained ambiguity between the uniform flow and the NUFF measurements. This ambiguity is manifested as a failure of the experimental data to fair smoothly and asymptotically to the measured uniform flow values of the force coefficients, e. g., C_N at ν = -4 deg, YP = -3. 16 in. (Fig. 6a) and C_m at ν = -4 deg, YP = -3. 16 in. (Fig. 6b). The calculated results from the NUFF, on the other hand, can be faired into the potential flow results in uniform flow with a minimum effort.

4.4.2 Comparison at M = 0.85

As might be expected from results in uniform flow (Fig. 3), the discrepancies in absolute values of calculated and experimental force coefficients were found to be even greater at M = 0. 85 than at M = 0. 5. On the incremental basis, however, the results are much closer to the M = 0. 5 results. Instead of the complete incremental comparison of calculated and measured coefficients given in Fig. 7 for M = 0.5, in Fig. 9 only two representative comparisons are given at M = 0.85. The best incremental comparison is for C_N versus ZP at YP = -3.16 in. (Fig. 9a), and the worst comparison is for C_m versus ZP at YP = -6.16 in. (Fig. 9b). A summary of the average discrepancy between theory and experiment at M = 0.85 is given in Fig. 10. At the higher Mach number the NUFF has a greater effect on the moment coefficients than on the force coefficients. The average discrepancies are 4 percent for force coefficients and 8 percent for moment coefficients. Also, the variation of the average discrepancies with the separation distance shows that sensitivity to mutual interference near the aircraft is much greater at the higher than at the lower Mach number. Discrepancies reach 10 percent at about 2 diameter separations for force coefficients and at about 3.5 diameters for moment coefficients. At the separation at which there is a 10- percent discrepancy in force coefficient, there is a 20- percent discrepancy in moment coefficient. The compressibility correction ranges from slightly beneficial to slightly non-beneficial.

4.5 TYPICAL STORE SEPARATION TRAJECTORIES

In order to demonstrate the viability of the trajectory/potential flow routine of Program C and to determine time of computation, calculations were performed for several typical separation trajectories. The solutions were obtained for a 250-lbm M-117 bomb launched at a 5000-ft altitude at M = 0.5 from Station No. 2 of an otherwise empty Triple Ejection Rack (TER) located on the left inboard pylon of an F-4C (Fig. 4). The bomb was assumed to be at a pitch attitude of -0.7 deg with respect to the free stream upon release from the carriage. Pitch and yaw damping rates were both set equal to -2.319 per radian (assuming the definitions of each rate to be based upon body length). Roll damping was neglected. The initial trajectory conditions resulted from an assumed ejector force of 1000 lb applied through a stroke of 0.2552 ft at 45 deg from the vertical in a direction passing through the center of gravity of the bomb. The trajectories shown in Fig. 11 commence at the end of the ejector stroke, i.e., at the instant the store separates from the TER. The translational displacements are computed with respect to a wind axes coordinate system where the origin is located at the store center of gravity at the instant of separation; the rotational displacements are with respect to the store body axes reference system.

In order to indicate the possible magnitude of perturbations to store motion caused by the nonuniform flow field, results of trajectory calculations for a store assumed ejected into both uniform and nonuniform flow fields are shown. For convenience, the nonuniform flow calculations were performed using absolute values of force coefficients obtained from the potential flow calculations, rather than by use of the incremental approach suggested in Section 4. 4. 1. This approach was considered compatible with the limited intent of demonstration of the program capability, independent of any considerations of accuracy. The most notable differences caused by flow nonuniformity are in the angular motion; the most significant effect on linear translation is in the x-direction.

The amplitudes of the pitch and yaw oscillations attenuate; the roll amplitude (not shown in Fig. 11) increases linearly with time in the absence of roll damping. It may be seen that even in a uniform flow field the instantaneous pitch and yaw motions occur about average amplitudes which are nonzero. The principal reason for this is that the existence of finite velocity components of the store in the Y- and Z-directions gives rise to effective angles of pitch and yaw different from the initial values of these angles. Hence, it is expected that these effective angles will tend to be mean values of the rotational excursions.

SECTION V
SUMMARY

Results of calculations may be summarized as follows:

1. A potential flow computer program based on vortex singularities, capable of accepting nonuniform flow boundary conditions but previously restricted to flows having a plane of symmetry, was modified to eliminate this restriction. The added capability did not alter any of the basic mathematics of the program but did effectively double the size of the system of simultaneous equations representing the flow for any given body.
2. In order to perform the expanded calculations within the limits of computer internal memory and desired computer execution time, the program was reorganized into three separate programs. The first two of these programs contain purely geometrical calculations and require execution only one time for a given

body geometry and Mach number. The third program performs calculations of vorticity distribution, pressure coefficient distribution, and force coefficients on the body and is executed one time for each set of boundary conditions (i.e., orientation and location in a nonuniform flow field). Once the initial geometric calculations are completed, potential flow solutions are generated by the third program in an average computer time of 17 sec for a vortex network composed of 312 horseshoe vortices.

3. The first program performs mainly the calculation of influence coefficients, and performs the computations in 10 min for a network of 312 vortices.

4. The second program performs mainly the inversion of the 312 rows by 312 columns coefficient matrix necessary to the potential flow solutions. Since the vortex network possessed geometric symmetry, even though in general there was no flow symmetry, the geometric matrices were partitioned, and the inversion then involved processing of only 156 rows by 156 columns matrices. A computer routine was devised which performs the inversion in 14 minutes.

5. In addition to reorganization of the structure of the program, additions were made to account for a proper base force on the vortex network, a linearized subsonic compressibility correction, and a skin-friction contribution. With these corrections, the force characteristics computed for a 312-vortex model of the M-117 bomb in a uniform flow were found to agree with wind tunnel measurements (except for drag) to within 10 percent at a Mach number of 0.5. To achieve this result, it was necessary to include in the vortex modeling of the bomb a number of vortices having the sole purpose of imposing a wakelike flow at the base of the bomb.

6. The force characteristics on the M-117 bomb in pitch and yaw were calculated by the potential flow program at 12 different locations in the disturbed flow field under an F-4C aircraft having an empty TER on the inboard pylon and a 370-gal external fuel tank on the outboard pylon. The absolute values of these force

coefficients were observed to differ from comparable wind tunnel measurements by up to 50 percent at a Mach number of 0. 5 and by greater amounts at a Mach number of 0. 85.

7. On the other hand, the incremental variations of the calculated coefficients in the nonuniform flow field were found to be in much closer agreement with measurements. The average discrepancy between calculated and measured incremental variations was from 4 to 6 percent at a Mach number of 0. 5. At a Mach number of 0. 85, the average discrepancy was from 4 to 6 percent for force coefficients and from 8 to 10 percent for force coefficients and from 8 to 10 percent for moment coefficients.

8. Discrepancies significantly greater than the average were observed at small separations of the bomb and aircraft, apparently because of neglect of mutual interference in the potential flow program. At a Mach number of 0. 5, a 10-percent discrepancy was reached at a separation distance of 1/2 the bomb diameter, while at a Mach number of 0. 85, the 10-percent discrepancy occurred at two diameters for force coefficients and 3. 5 diameters for moment coefficients.

9. In view of the relatively accurate predictions of incremental force coefficient behavior (as opposed to absolute values), it was apparent that a hybrid method of force coefficient determination in a disturbed flow field was possible. Force coefficients measured on a store at a limited number of points far from a parent aircraft could apparently be incremented at other points in the disturbed flow field by reference to potential flow solutions.

10. The linearized compressibility correction was found to have very little influence in determination of the incremental variation of force coefficients.

11. A six-degree-of-freedom trajectory routine was appended to the third program, using the same basic equations of motion as in the AEDC/PWT CTS. In order to minimize the number of time steps required for a given accuracy of trajectory (and, hence, the

number of potential flow solutions), a more sophisticated numerical integration procedure (fourth-order Runge-Kutta) was used. A limited number of M-117/F-4C trajectories were computed with non-uniform and uniform flow force coefficients, respectively, to display the magnitude of effects of the disturbed flow field.

REFERENCES

1. MacDermott, W. N. and Johnson, P. W. "Calculation for Forces on Aircraft Stores Located in Disturbed Flow Fields for Application in Store Separation Prediction." AEDC-TR-71-186 (AD733325), November 1971.

2. Ward, G. N. Linearized Theory of Steady High-Speed Flow. Cambridge Universtiy Press, Cambridge, 1955.

3. Rubbert, Paul E. "Theoretical Characteristics of Arbitrary Wings by a Non-Planar Vortex Lattice Method." Boeing Company, Document No. D6-9244, February 1964.

4. Christopher, J. P. and Carleton, W. E. "Captive-Trajectory Store-Separation System of the AEDC-PWT 4-Foot Transonic Tunnel." AEDC-TR-68-200 (AD839743), September 1968.

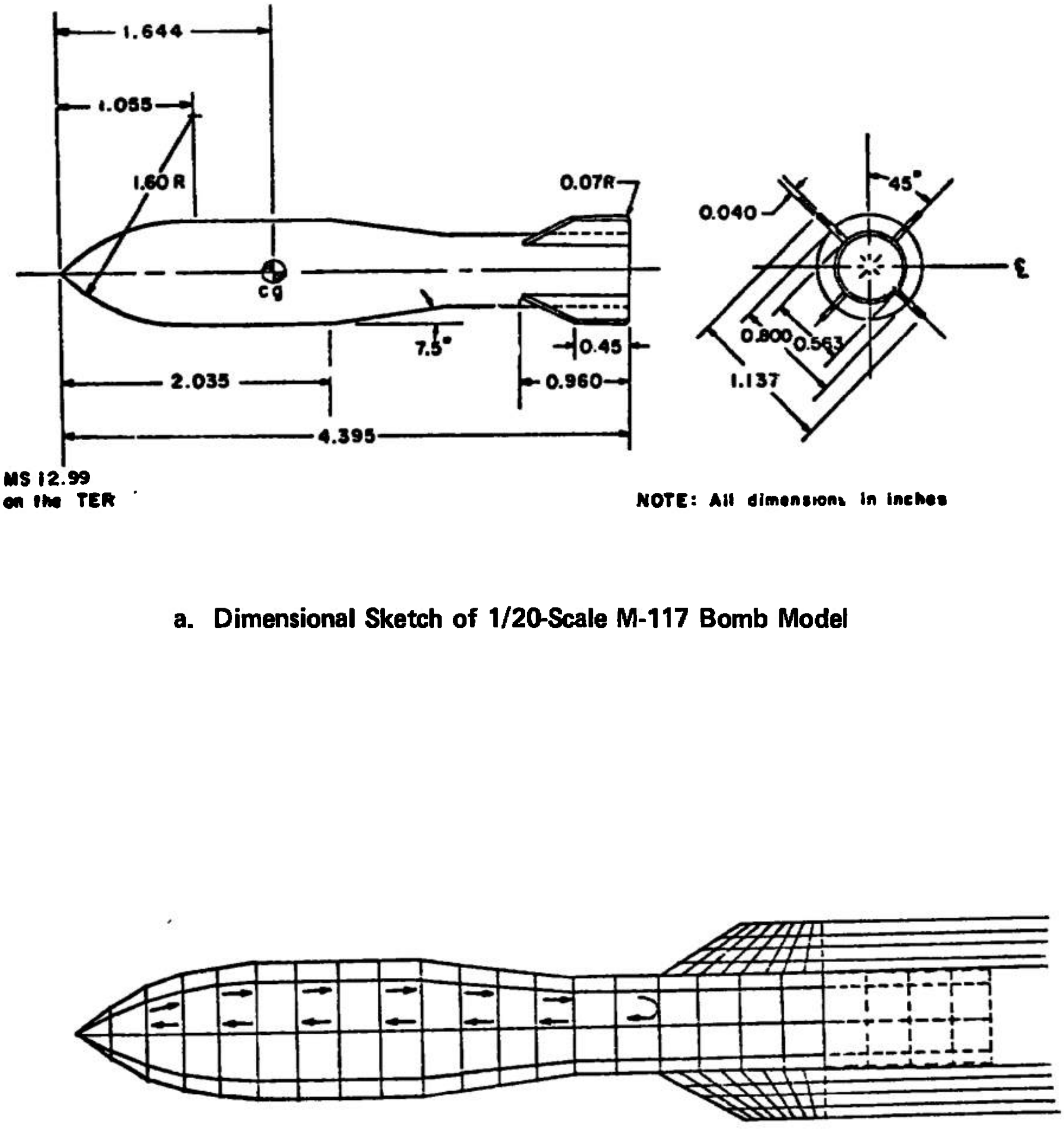
5. Thelander, J. A. "Aircraft Motion Analysis." FDL-TDR-64-70 (AD617354), March 1965.

6. Davis, Ronald E. "Flow Field Characteristics Beneath the F-4C Aircraft at Mach Numbers 0.50 and 0.85." AEDC-TR-70-8 (AD864702), February 1970.

APPENDIXES

I. ILLUSTRATIONS

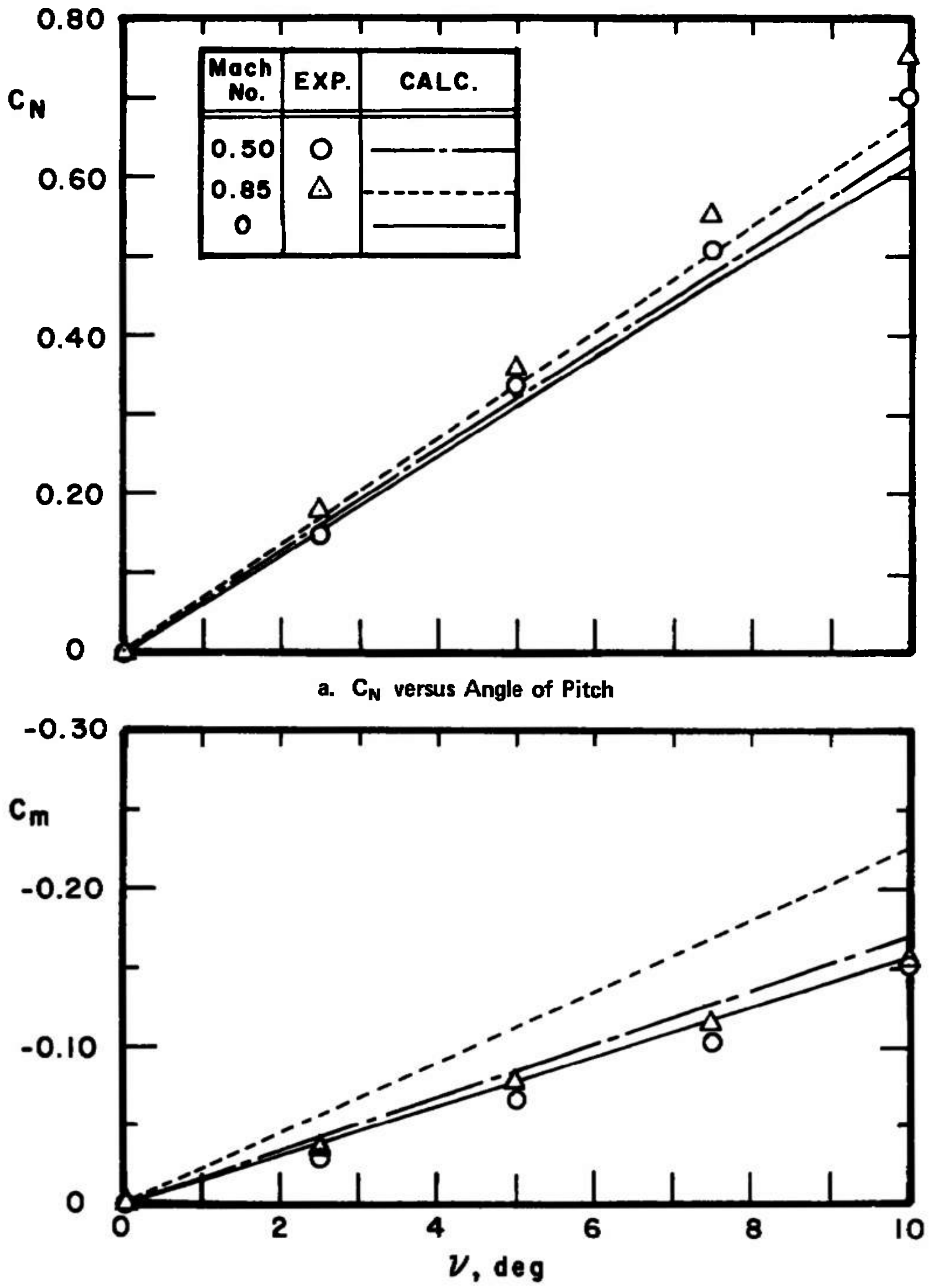
II. USER'S GUIDE TO COMPUTER PROGRAM



a. Dimensional Sketch of 1/20-Scale M-117 Bomb Model

156-Vortex Model

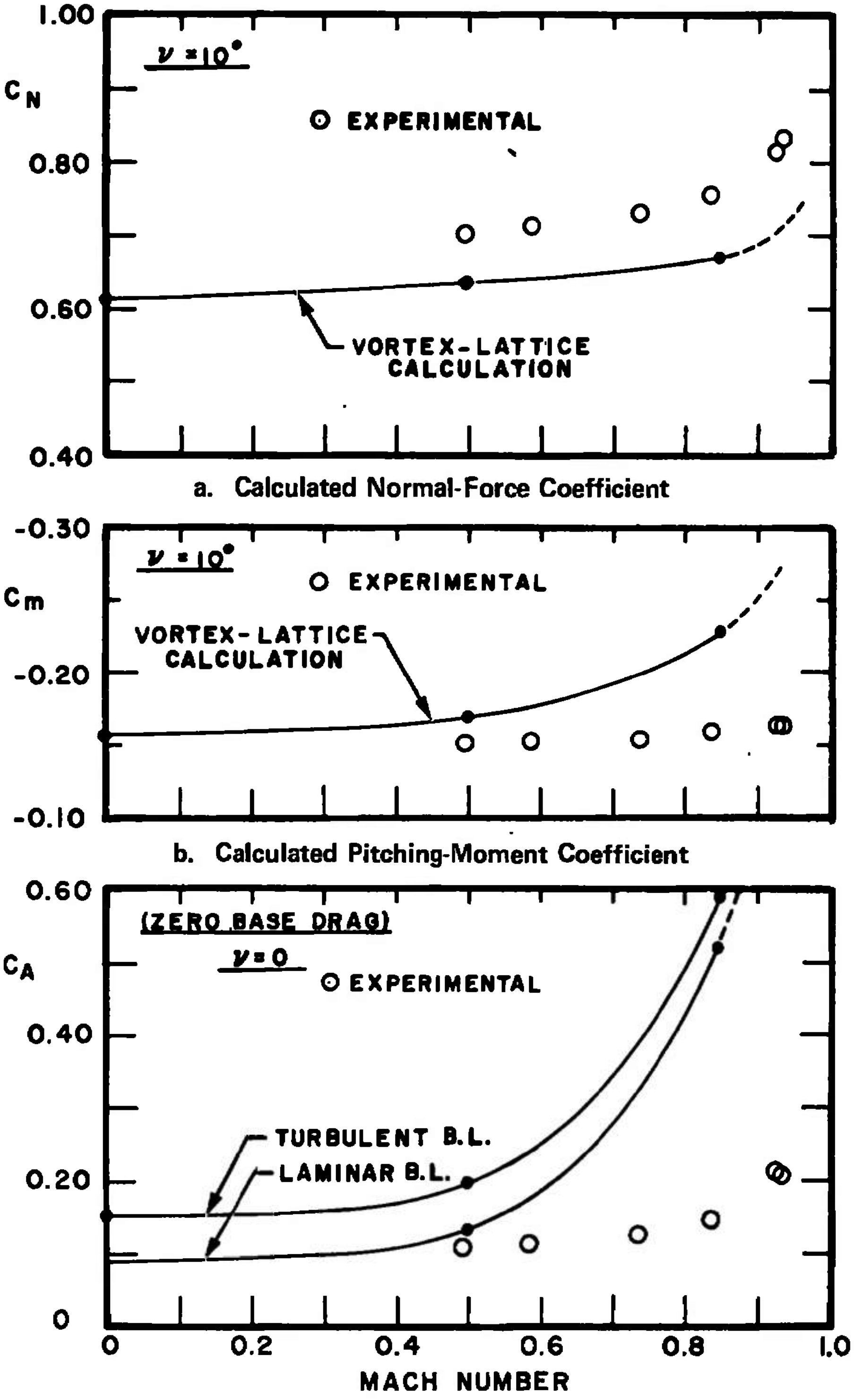
b. 312 Vortex Model for Approximation of M-117 Bomb
Fig. 1 Shape of M-117 Bomb



a. C_N versus Angle of Pitch

b. C_m versus Angle of Pitch

Fig. 2 Comparison of Calculated and Experimental Force Coefficients on M-117 Bomb, Uniform Flow



a. Calculated Normal-Force Coefficient

b. Calculated Pitching-Moment Coefficient

c. Calculated Axial-Force Coefficient

Fig. 3 Effect of Mach Number on Calculated and Experimental Force Coefficients, Uniform Flow

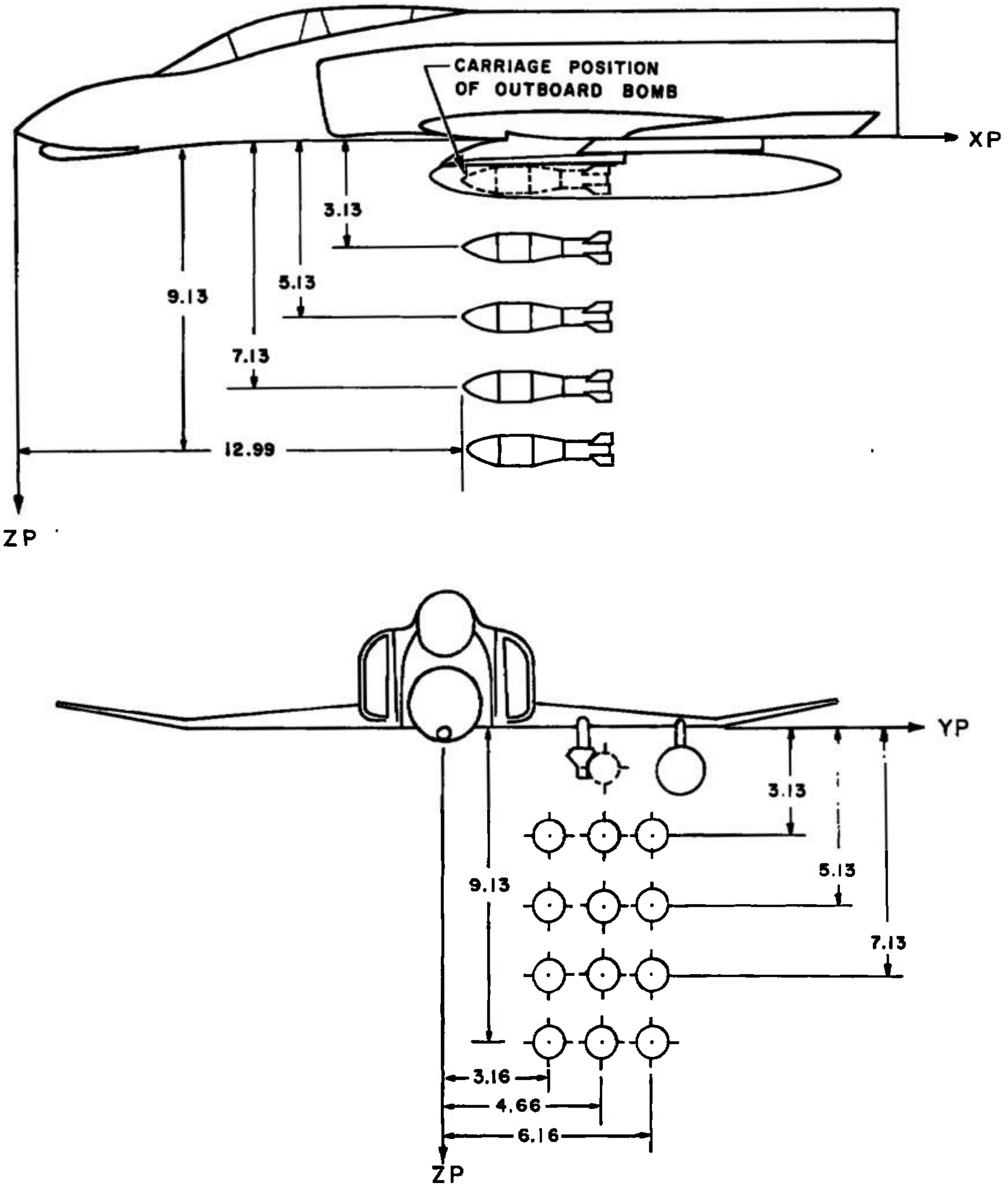


Fig. 4 Locations of Bomb under F-4C Parent Aircraft at Which Calculated and Measured Forces Were Compared

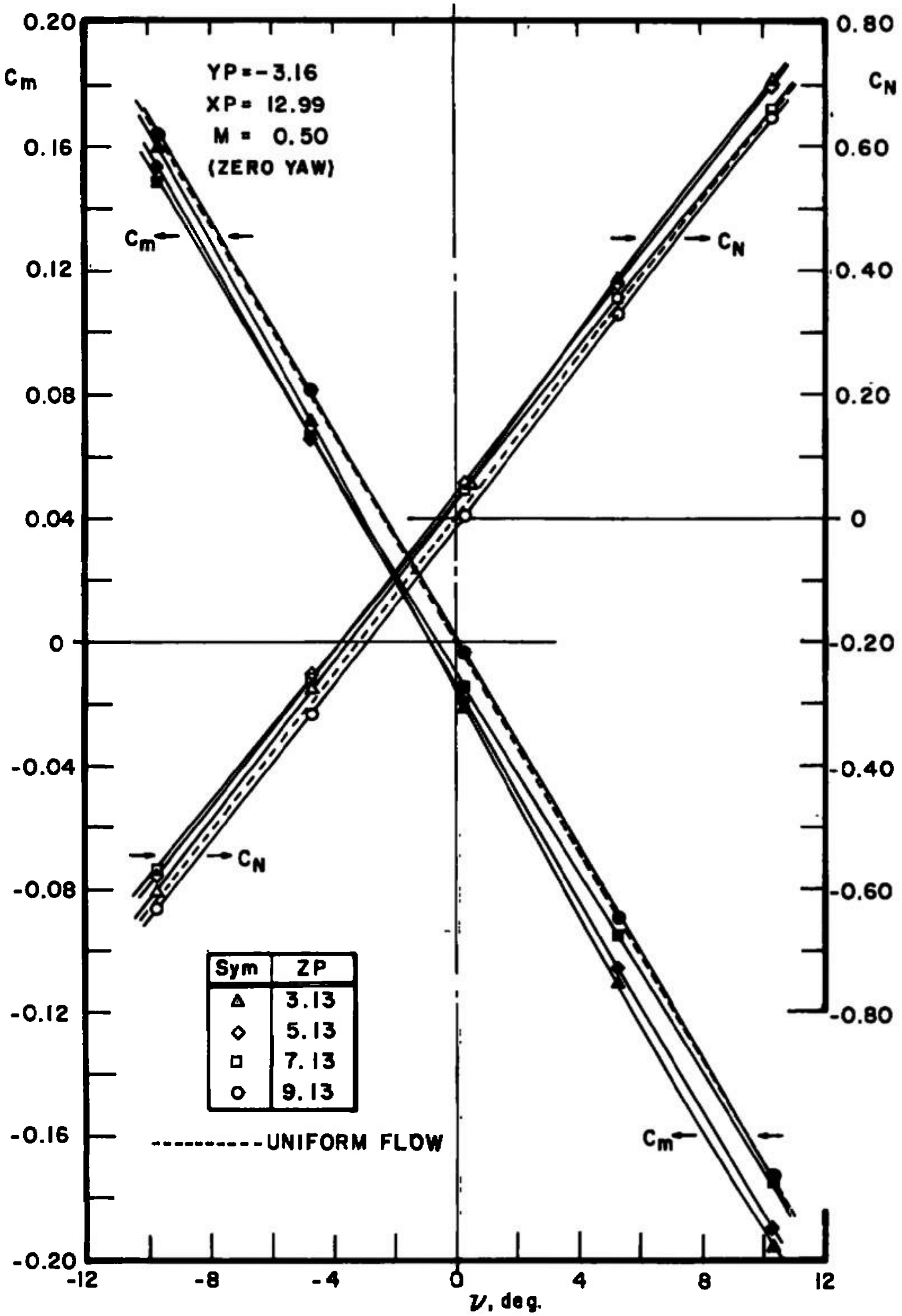
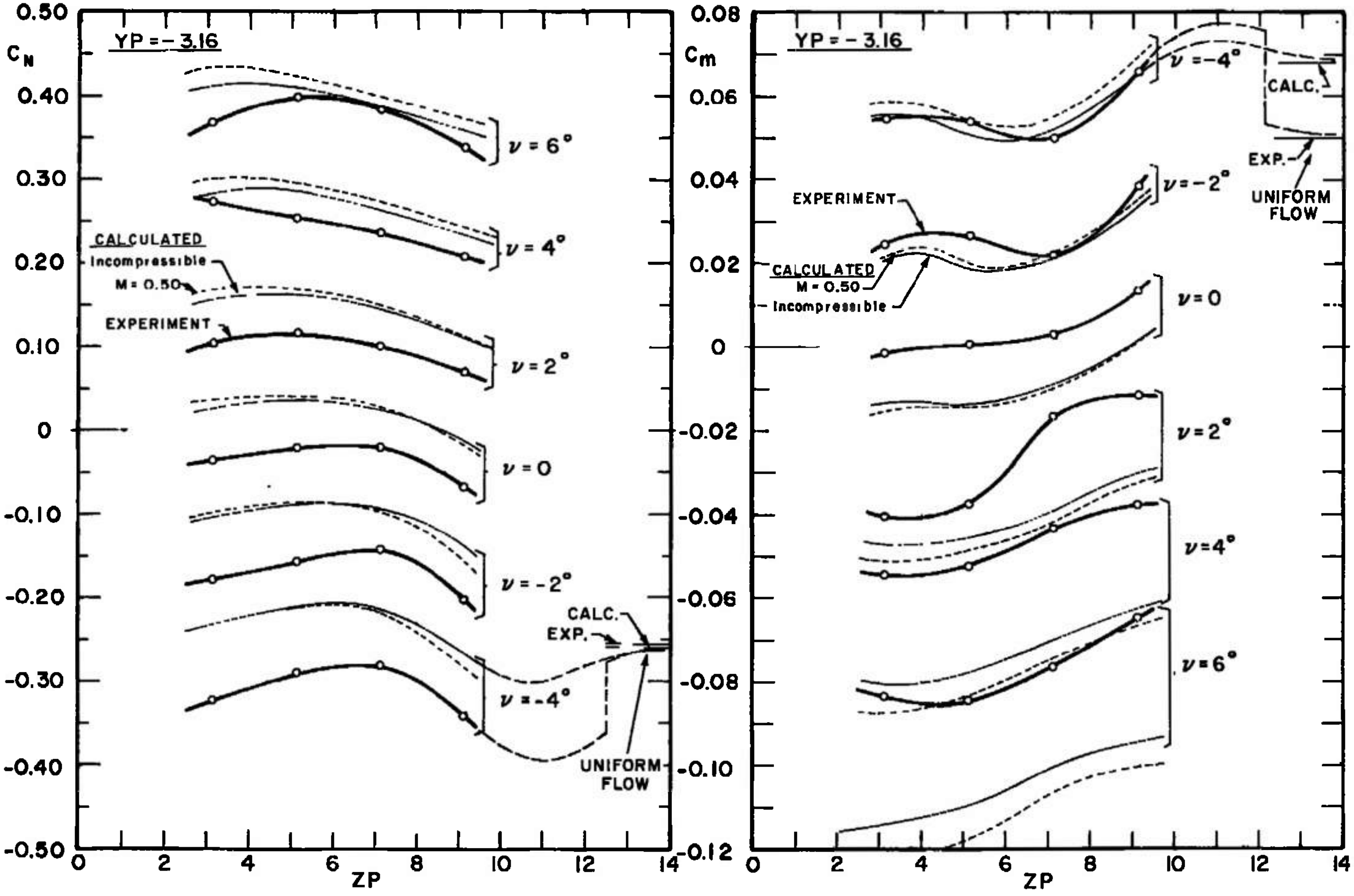


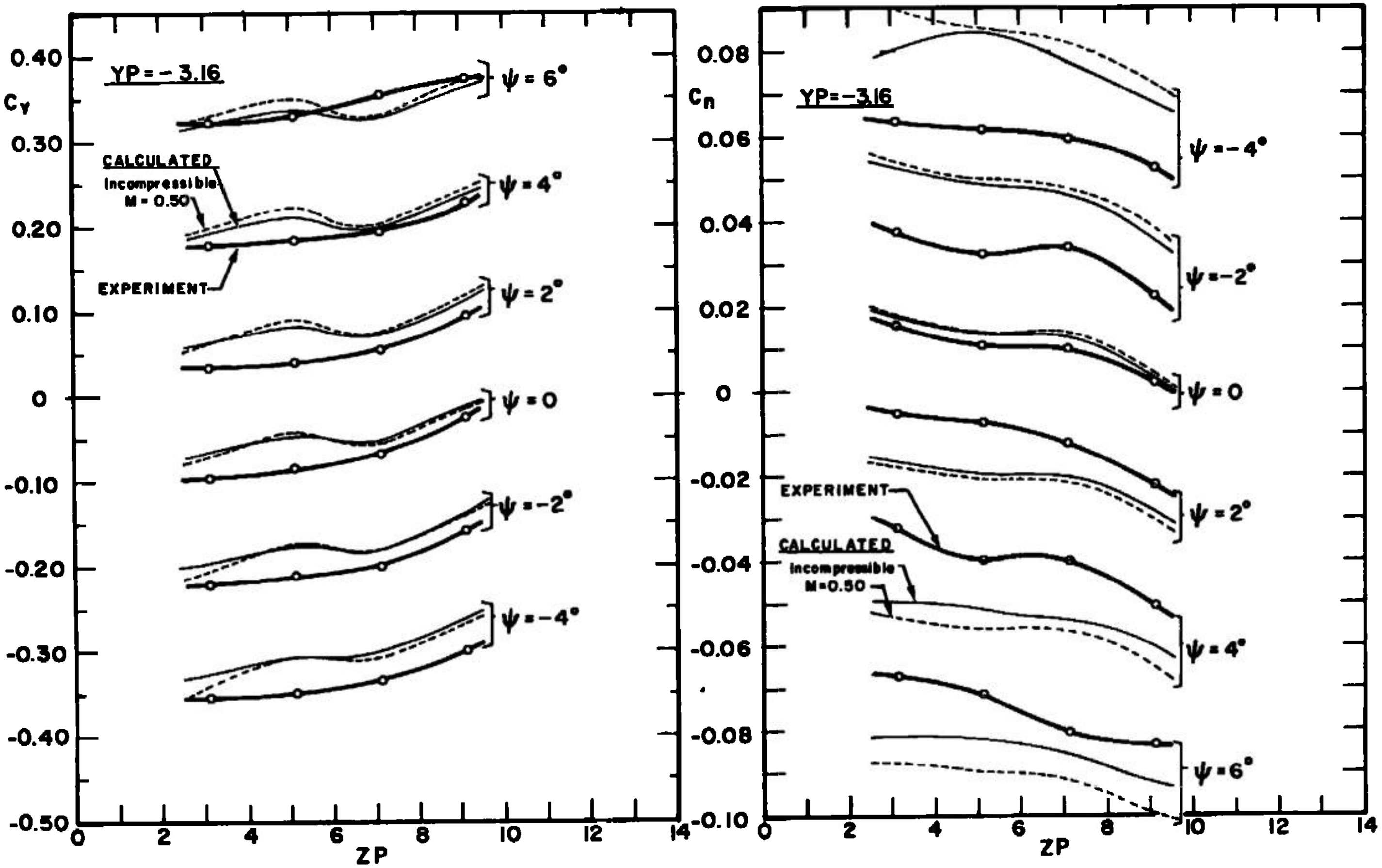
Fig. 5 Calculated Variation of C_N and C_m with Pitch Angle in F-4C Flow Field, YP = -3.16, XP = 12.99, M = 0.50



a. C_N versus ZP, Constant Pitch, Zero Yaw

b. C_m versus ZP, Constant Pitch, Zero Yaw

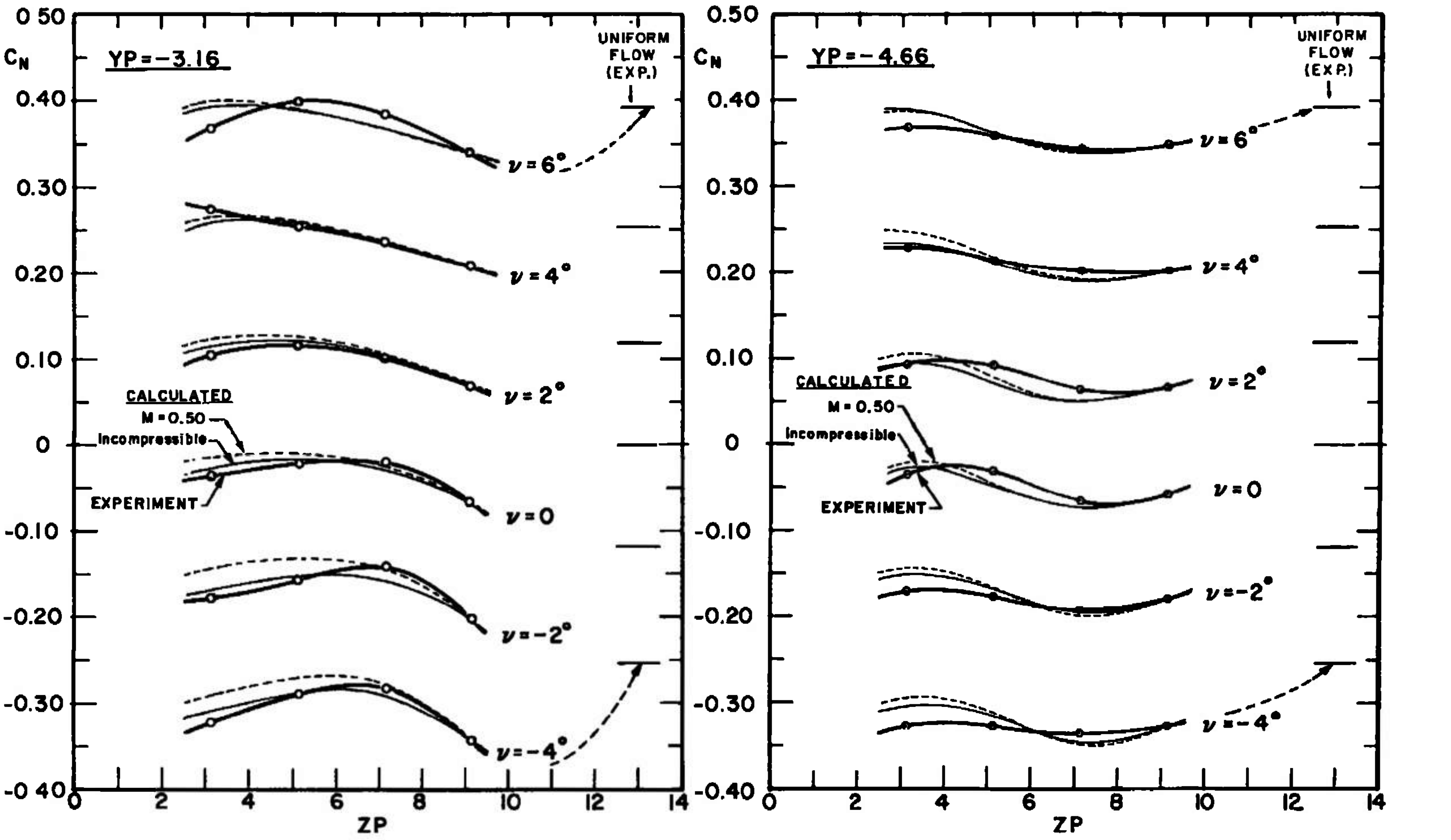
Fig. 6 Comparison of Absolute Values of Calculated and Experimental Force Coefficients in F-4C Flow Field, M = 0.5, YP = -3.16



c. C_Y versus ZP, Constant Yaw, Zero Pitch

d. C_n versus ZP, Constant Yaw, Zero Pitch

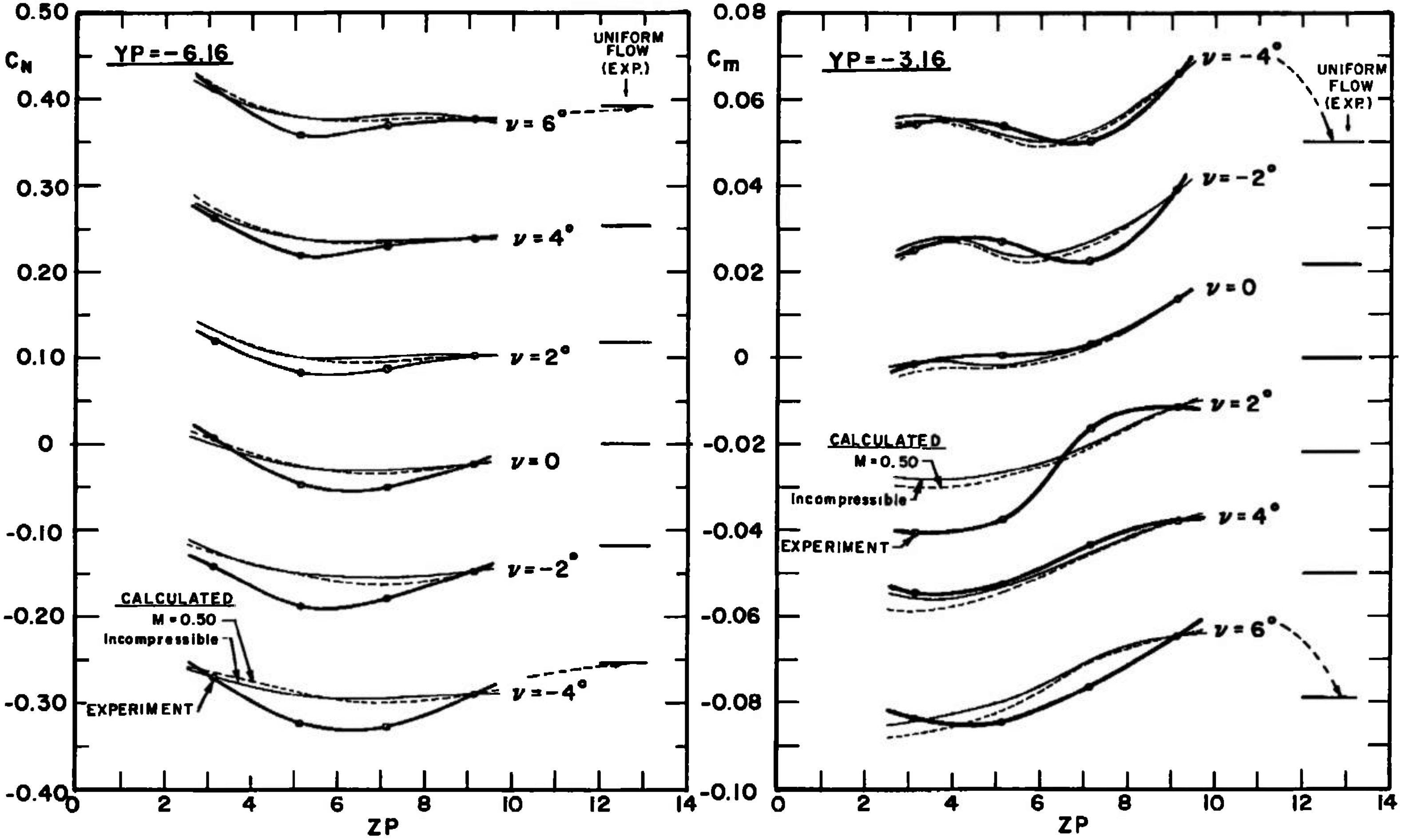
Fig. 6 Concluded



a. C_N versus ZP, Constant Pitch, Zero Yaw, YP = -3.16

b. C_N versus ZP, Constant Pitch, Zero Yaw, YP = -4.66

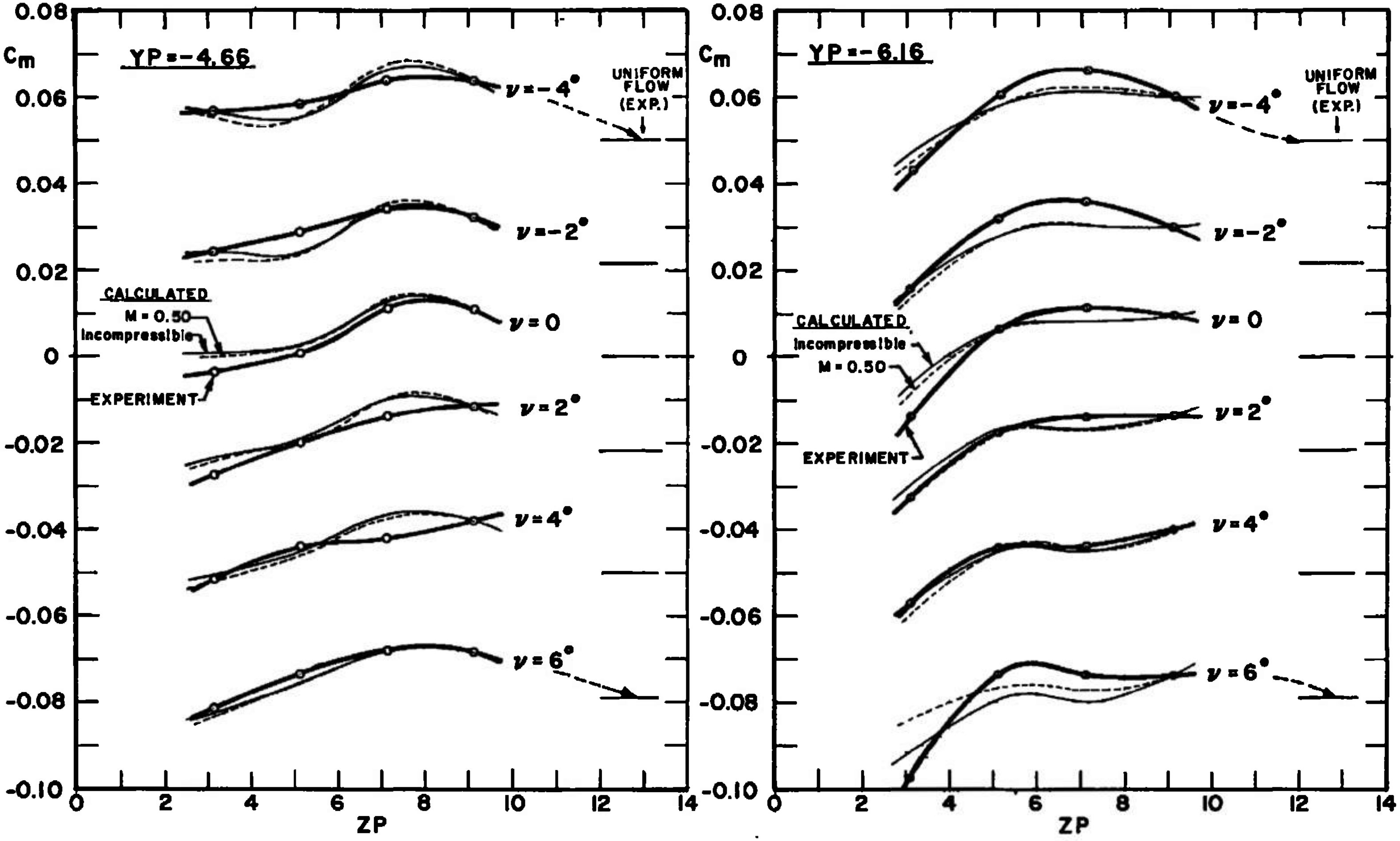
Fig. 7 Comparison of Incremental Variation of Calculated and Experimental Force Coefficients in F-4C Flow Field, M = 0.5



c. C_N versus ZP, Constant Pitch, Zero Yaw, YP = -6.16

d. C_m versus ZP, Constant Pitch, Zero Yaw, YP = -3.16

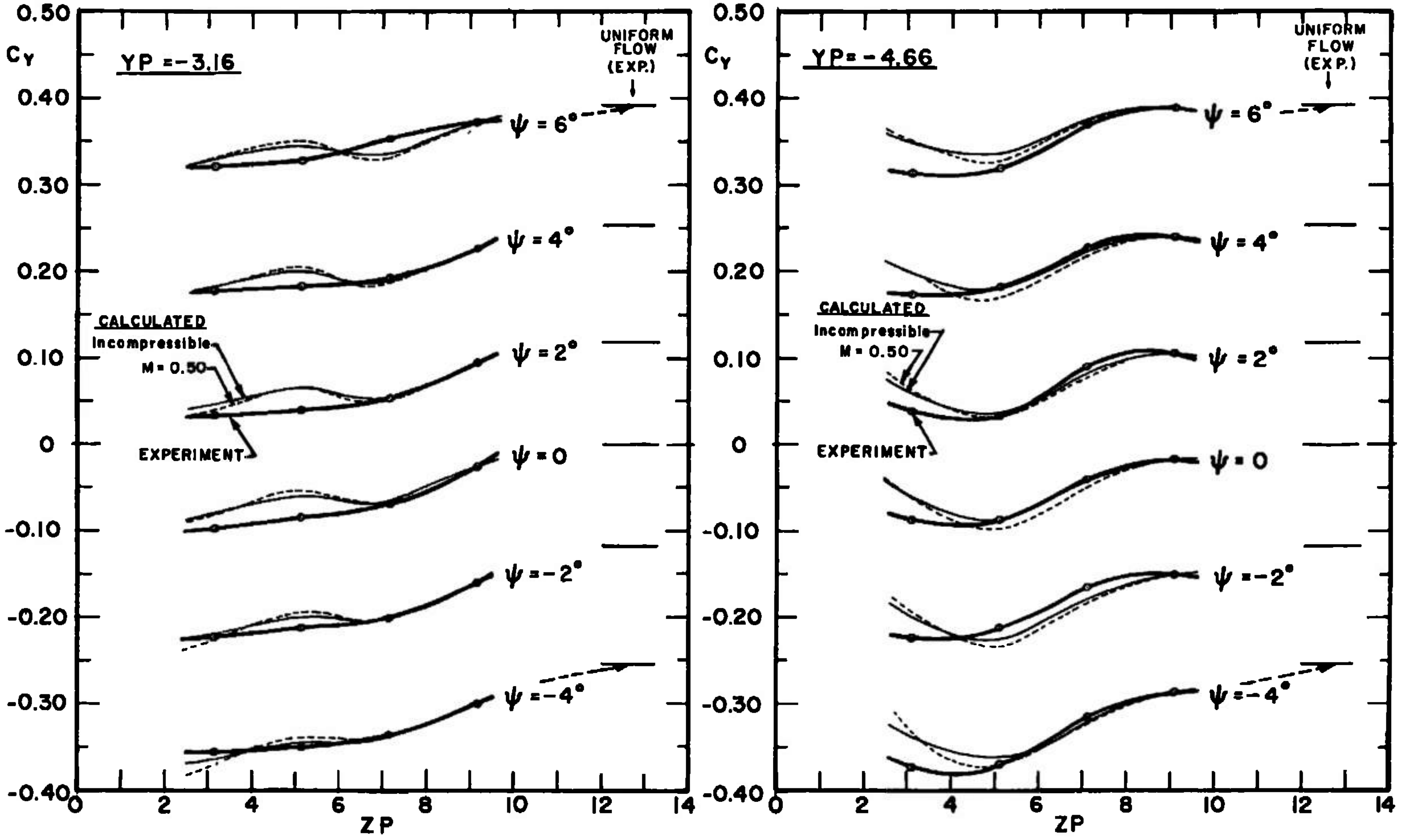
Fig. 7 Continued



e. C_m versus ZP, Constant Pitch, Zero Yaw, YP = -4.66

f. C_m versus ZP, Constant Pitch, Zero Yaw, YP = -6.16

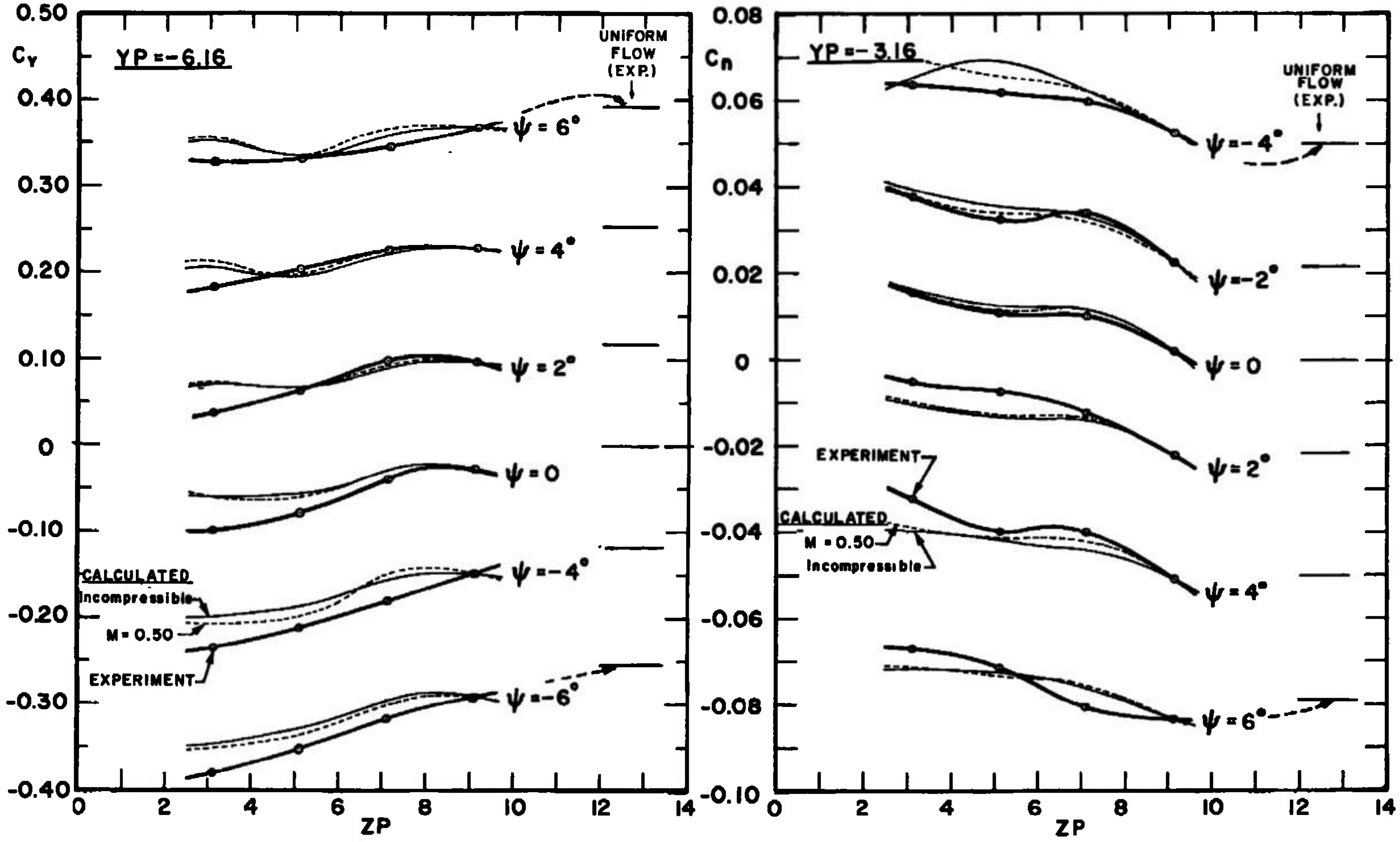
Fig. 7 Continued



g. C_Y versus ZP, Constant Yaw, Zero Pitch, YP = -3.16

h. C_Y versus ZP, Constant Yaw, Zero Pitch, YP = -4.66

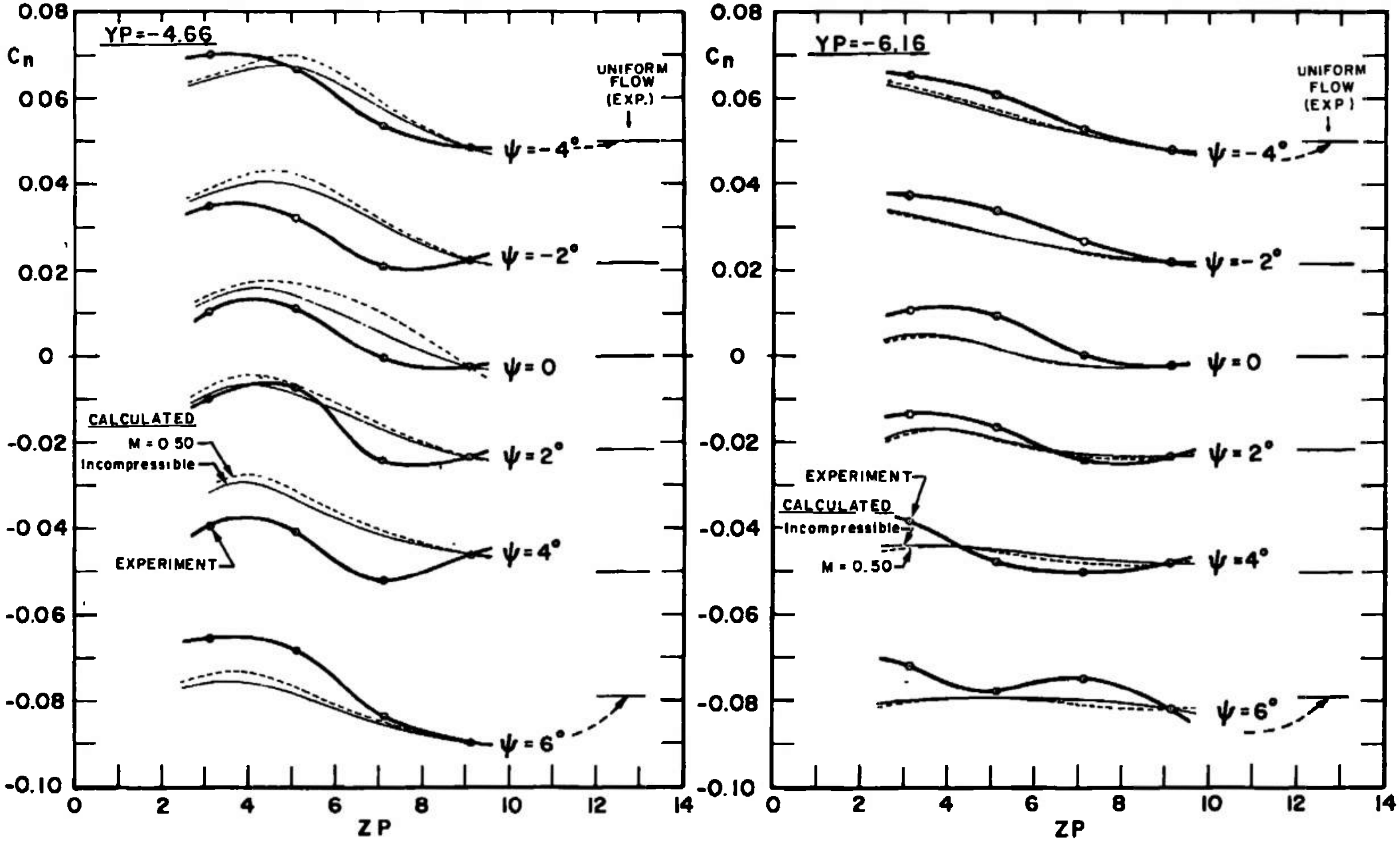
Fig. 7 Continued



i. C_Y versus ZP, Constant Yaw, Zero Pitch, YP = -6.16

j. C_n versus ZP, Constant Yaw, Zero Pitch, YP = -3.16

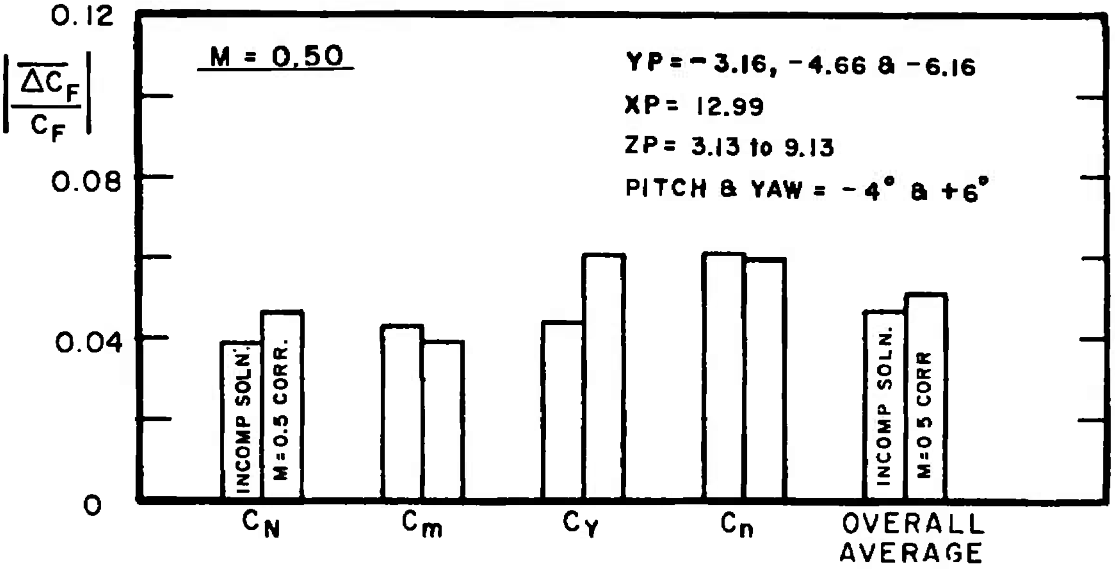
Fig. 7 Continued



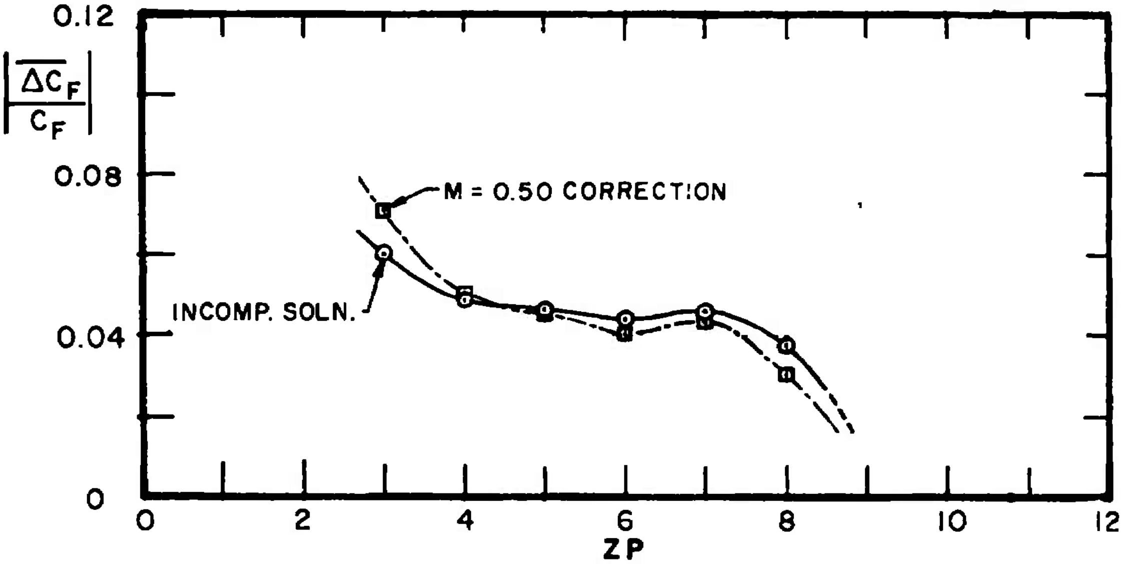
k. C_n versus ZP, Constant Yaw, Zero Pitch, YP = -4.66

l. C_n versus ZP, Constant Yaw, Zero Pitch, YP = -6.16

Fig. 7 Concluded

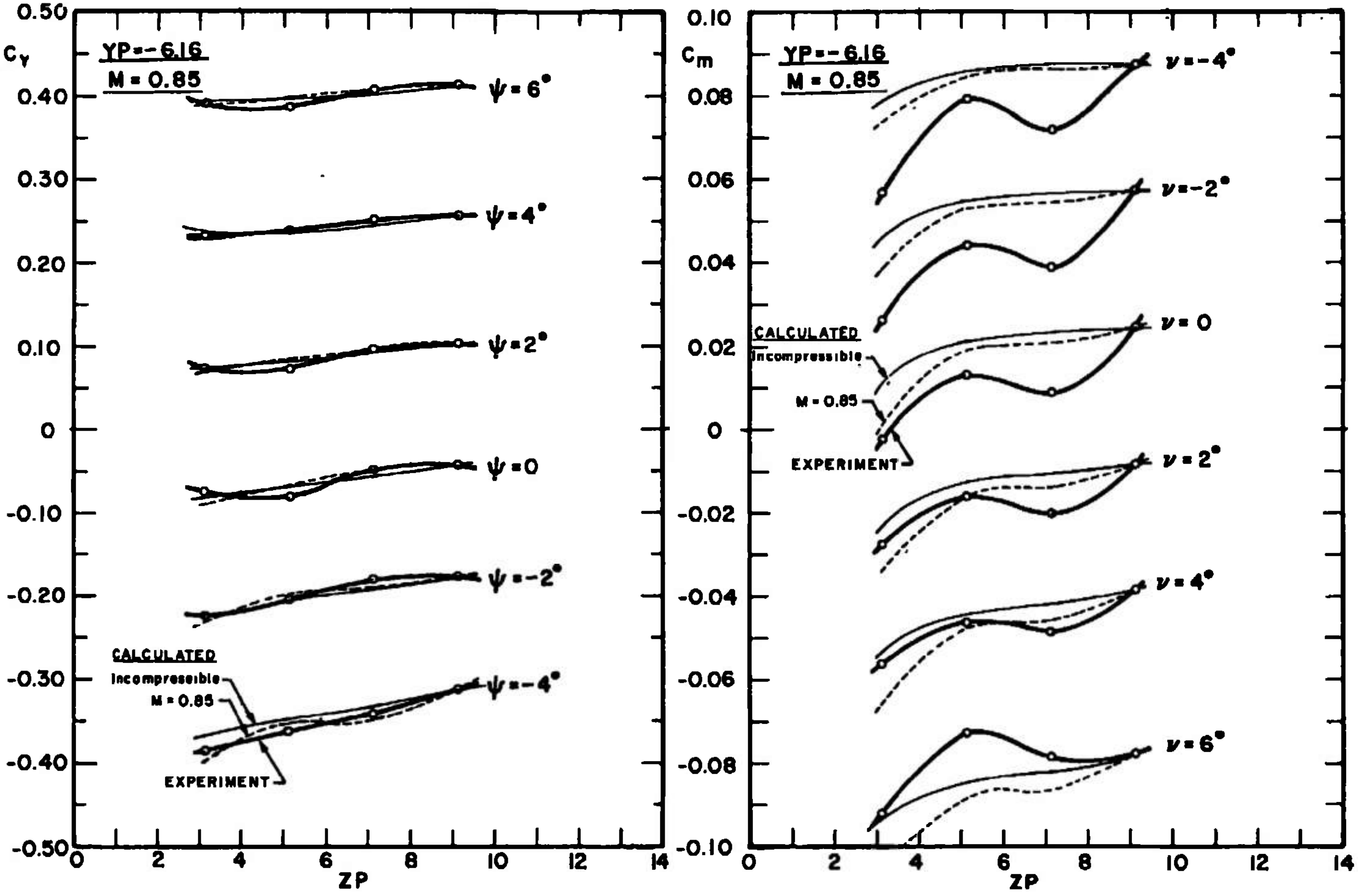


a. Average Discrepancy in C_N, C_m, C_Y, C_n, and Overall



b. Variation of Average Theoretical/Experimental Discrepancy with ZP

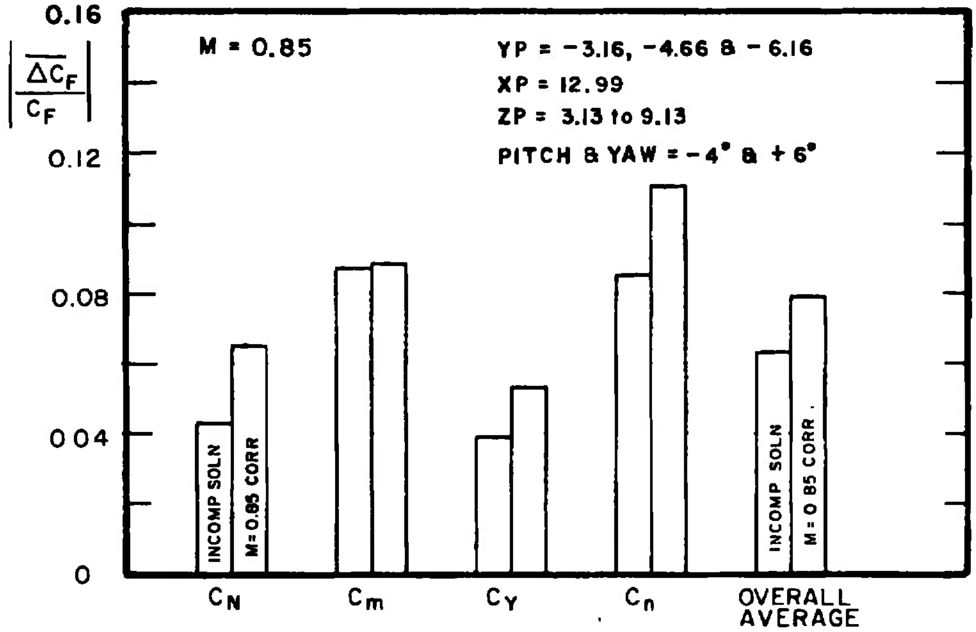
Fig. 8 Average Discrepancy Between Calculated and Experimental Force Coefficient, Incremental Basis, Pitch and Yaw = -4 deg and 6 deg, M = 0.5



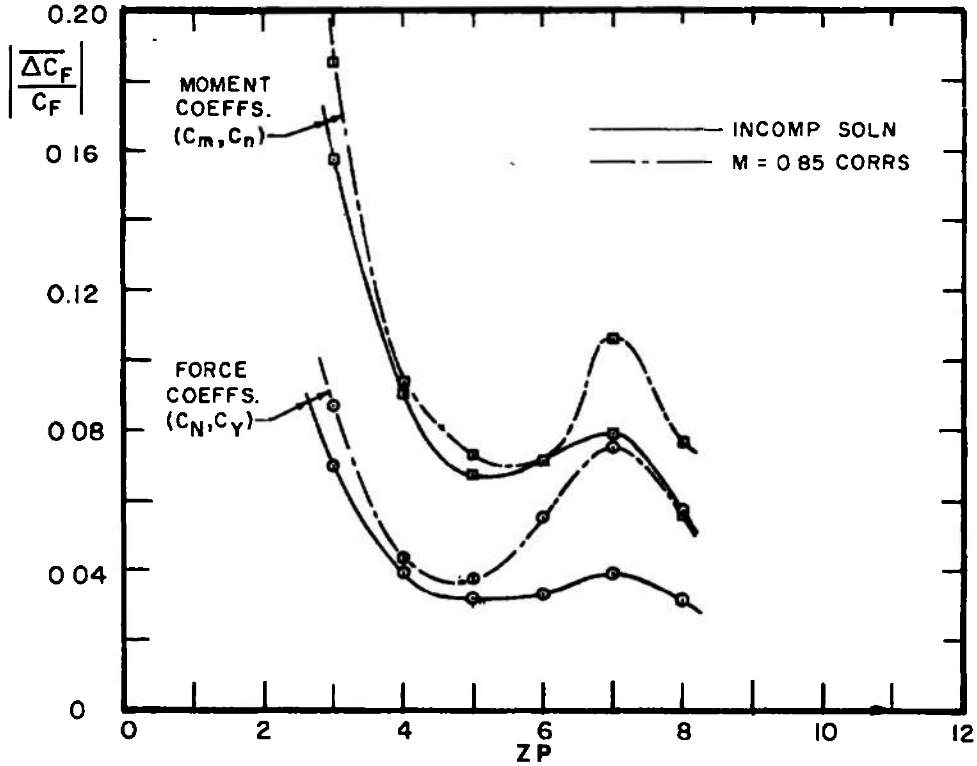
a. Best Comparison, C_Y versus ZP, Constant Yaw, Zero Pitch, YP = -6.16

b. Worst Comparison, C_m versus ZP, Constant Pitch, Zero Yaw, YP = -6.16

Fig. 9 Comparison of Incremental Variation of Calculated and Experimental Force Coefficients in F-4C Flow Field, M = 0.85



a. Average Discrepancy in C_N, C_m, C_Y, C_n, and Overall



b. Variation of Average Discrepancy with ZP

Fig. 10 Average Discrepancy Between Calculated and Experimental Force Coefficients, Incremental Basis, Pitch and Yaw = -4 deg and 6 deg, M = 0.85

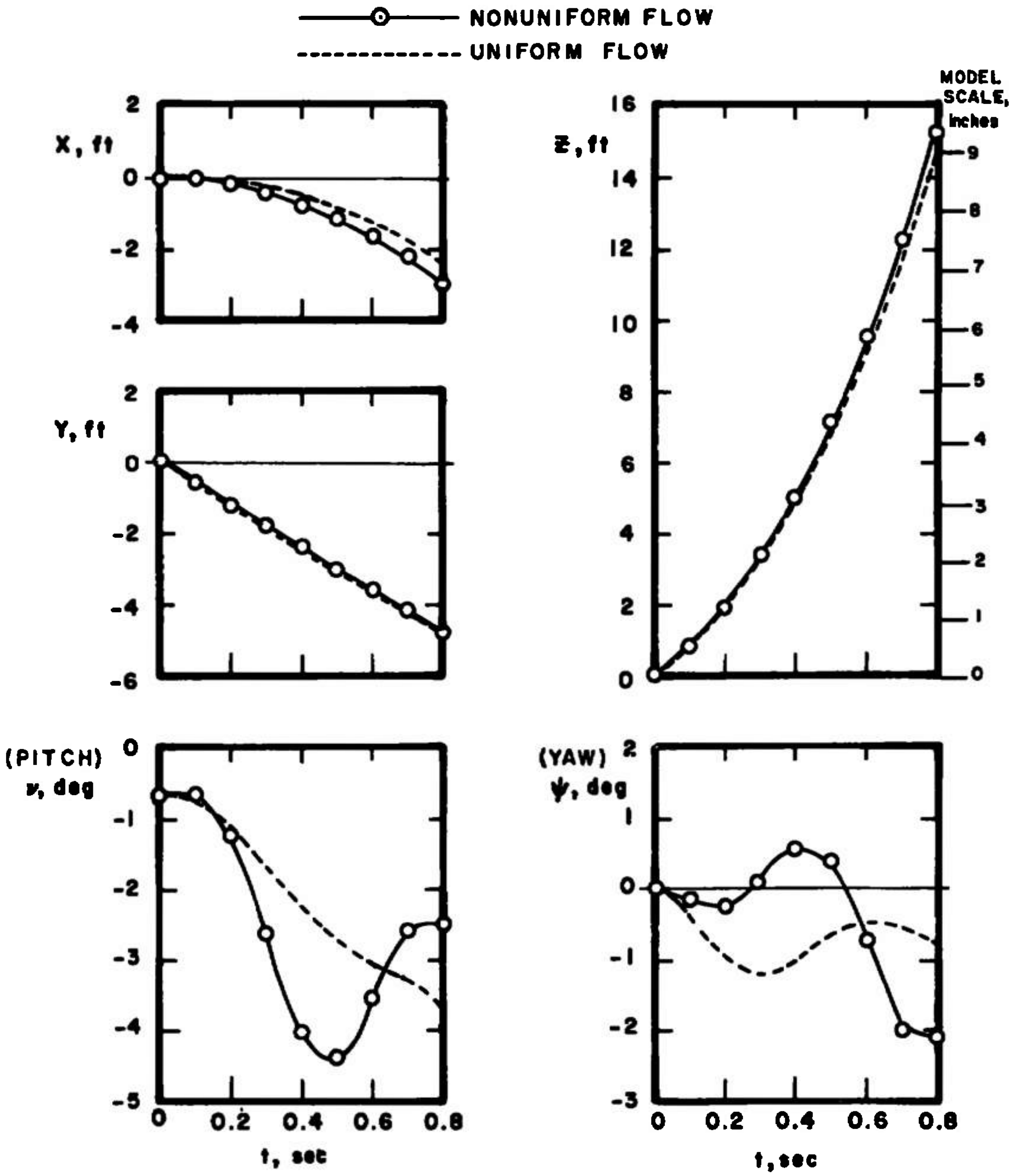


Fig. 11 Comparison of Separation Trajectories with Uniform Flow Force Coefficients and Nonuniform Flow Force Coefficients, F-4C Aircraft, 250-lbm M-117 Bomb, on No. 2 TER Station, Inboard Pylon, M = 0.50, 5000-ft Altitude, Pitch Angle of A/C = 0.30 deg, 1000-lb Ejector Force at 45 deg to Vertical

APPENDIX II
USER'S GUIDE TO COMPUTER PROGRAM

A user's guide for application of the vortex-lattice and store trajectory computer program is presented. This includes a listing of the source program, together with a description of the input data and other variables for which values must be furnished by the user. A diagram of general program structure and subroutine function is given in Fig. II-1. In order to assist the user in becoming familiar with the operational aspects of the program, input and output data are displayed for a sample run.

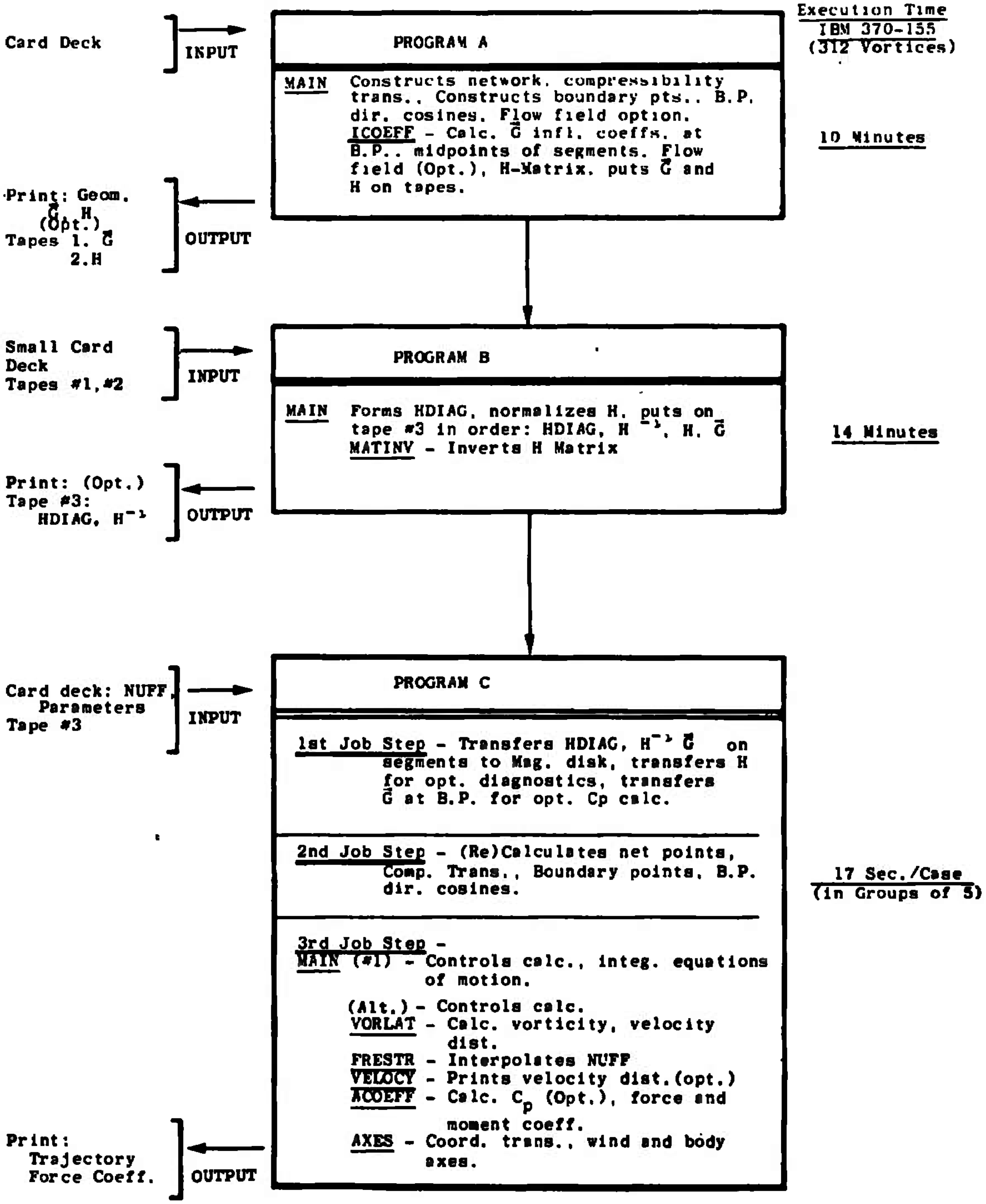
The program is coded in the FORTRAN IV Language for execution on the IBM System/360 or 370 computer. The values for the variables in the program which must be supplied by the user are provided in the following forms: inputs from punched card data sets, magnetic tape units, and disk storage units; arithmetic expressions; and maximum values of subscripts in DIMENSION statements.

The unit of length for all the variables defined in the MAIN program of Program C is feet. An exception to this occurs in the version of the MAIN program of Program C, which is used when trajectories are not computed. The units of the coordinates of the point of rotation of the store (denoted by XORIG, YORIG, and ZORIG) are inches. For all other variables in Programs A and C (the unit of length does not enter into Program B), length is prescribed in inches. The reason for this distinction is that a convenient unit for displacement of a full-scale store along a trajectory is feet, while a convenient unit for a model store and its associated NUFF is inches.

The sample run assumes that the store being analyzed is an M-117 bomb modeled with 156 vortices on each side of the x-z plane of geometrical symmetry, and a uniform flow field. (This letter assumption has been introduced to avoid the necessity of displaying NUFF input data below.) The store geometry and the vortex lattice used to model the store are shown in Fig. 1. For the sake of brevity, only a few representative values of output data are presented; these should be adequate for the user to verify his execution of the program.

VORTEX-LATTICE PROGRAM

(With Trajectory Option)



MAIN Constructs network, compressibility trans., Constructs boundary pts., B.P. dir. cosines, Flow field option.
ICOEFF - Calc. $\vec{G}$ infl. coeffs. at B.P., midpoints of segments. Flow field (Opt.), H-Matrix, puts $\vec{G}$ and H on tapes.

MAIN Forms HDIAG, normalizes H, puts on tape #3 in order: HDIAG, H^{-1}, H, $\vec{G}$
MATINV - Inverts H Matrix

1st Job Step - Transfers HDIAG, H^{-1} $\vec{G}$ on segments to Mag. disk, transfers H for opt. diagnostics, transfers $\vec{G}$ at B.P. for opt. Cp calc.

2nd Job Step - (Re)Calculates net points, Comp. Trans., Boundary points, B.P. dir. cosines.

3rd Job Step -
MAIN (#1) - Controls calc., integ. equations of motion.
(Alt.) - Controls calc.
VORLAT - Calc. vorticity, velocity dist.
FRESTR - Interpolates NUFF
VELOCY - Prints velocity dist. (opt.)
ACOEFF - Calc. C_p (Opt.), force and moment coeff.
AXES - Coord. trans., wind and body axes.

Fig. II-1 Diagram of Structure of Vortex-Lattice Program with Trajectory Option

2.1 USER'S GUIDE TO PROGRAM A

Input Data

The following data are read from punched cards. (The first card contains arbitrary descriptive information furnished by the user.)

AMACH — Free-stream Mach number

NWP — Number of wing parts (a wing part is defined in Ref. 3, p. 12

NCHORD(IW) — Number of vortex network points on wing part IW in the cordwise direction (Ref. 3, p. 10)

NSPAN(IW) — Number of network points in the spanwise direction

MTIP = 0 — Contiguous wing part tips (e. g., closed body)
= 1 — Terminal wing part tips (e. g., wing, tail, fins)

MROOT = 0 — Contiguous wing part roots (e. g., closed body)
= 1 — Terminal wing part roots (e. g., vertical tail, fins)

NSYM = 1 — Symmetry about x-z plane prevails (uniform flow field and no yaw or roll)
= 2 — Without symmetry about x-z plane (nonuniform flow field or yaw or roll)

Influence coefficients can be optionally computed at as many as four sets of points. The input which specifies whether or not the calculations are to be performed at each of the sets is IVEL(ISOLVE). When ISOLVE = 1, the set of points is the boundary points; when ISOLVE = 2, the set of points is the midpoints of spanwise vortex segements; when ISOLVE = 3, the set of points is the midpoints of the chordwise vortex segments, and when ISOLVE = 4, the points are arbitrarily located off the surface of the planform.

IVEL(ISOLVE) = 0 — Do not compute $\vec{G}$
= 1 — Compute $\vec{G}$

IPRG = 0 — Do not print $\vec{G}$
= 1 — Print $\vec{G}$

IPRH = 0 — Do not print H
= 1 — Print H

EPS, EPSR — Magnitude tests used in SUBROUTINE ICOEFF (Ref. 3, p. 26.) The criteria used to specify values of magnitude tests in the analysis reported herein are as follows: (1) that EPSR be an order of magnitude less than the smallest of the distances of boundary points to neighboring vortex segments, and (2) that EPS be an order of magnitude greater than the square of EPSR.

If velocity calculations are to be made at points in the flow field off the surface of the bomb (this requires specifying IVEL(4) = 1), the coordinates of these points (denoted by the names XFLOWF, YFLOWF, and ZFLOWF) are read from punched cards.

Variables which the user must define by FORTRAN arithmetic expressions are (1) the cordinates of the network points (XNET, YNET, and ZNET) and (2) the direction cosines (AX, AY, and AZ) of the vortices extending downstream from trailing edge network points. An aerodynamic planform can not be represented by a vortex network in an arbitrary manner. Guidelines for proper selection of the coordinates of network points are given in Section 5. 1 of Ref. 1 and in Section VIII of Ref. 3. Considerations involved in specifying the trailing vortex direction cosines are discussed in Section VIII. F of Ref. 3. In performing the calculations discussed in this report, the trailing vortices were assumed to be parallel to the axis of symmetry.

Dimensions of Arrays

The values of the dimensions of certain variables depend upon the detailed manner in which the aerodynamic planform is modeled with vortex networks. These variables, and the corresponding dimensions are as follows:

NCHORD(NWP), NSPAN(NWP), NSCORD(NWP),

NVOR(NWP), NUC(NWP, 4), NUS(NWP, 4), NVEL(NWP, 4),

BDC(NVORT, 3), AX(NWP), AY(NWP), AZ(NWP),

MTIP(NWP), MROOT(NWP), H(NVORT), G(NVORT, 3)

where NVORT is the total number of vortices on half of a planform (i. e., on one side of the x-z plane of symmetry).

The dimensions of the first, second, and third subscripts, respectively, of the network point coordinates XNET, YNET, and ZNET are (1) the maximum value of NCHORD(IW), (2) the maximum value of NSPAN(IW), and (3) NWP. The dimensions of the first, second, and third subscripts, respectively, of boundary point coordinates AKSI, ETA, and ZETA are (1) the maximum of NCHORD(IW)-1, (2) the maximum of NSPAN(IW)-1, and (3) NWP.

The dimensions of the first subscripts of DIC and EIC in SUBROUTINE ICOEFF are each equal to the maximum value of NCHORD(IW).

PROGRAM A
INPUT DATA FROM PUNCHED CARDS
SAMPLE RUN

```
 G AND H MATRICES FOR AN M-117 BOMB AT M=0.0 ARE COMPUTED FOR SAMPLE RUN
 0.0

    3
   24    5    9    5    9    5

    0    0    1    1    1    1
    2

    1    1    1    0
    1    1

    1.E-05    1.E-03
```

```
C      PROGRAM A
C      G AND H MATRICES ARE COMPUTED IN THIS PROGRAM
       COMMON/CNET  /      NWP             , NIWP            ,
     1    NCHORD(3)     , NSPAN (3)     , NSCORD(3)     , NVOR   (3)    ,
     2    NUC    (3,4)  , NUS    (3,4)  , NVEL   (3,4)  , NUT       (4)
       COMMON/CCOORD/      XNET (24,5,3), YNET (24,5,3), ZNET (24,5,3),
     1                     AKSI (23,4,3), ETA  (23,4,3), ZETA (23,4,3),
     2                     XFLOWF( 1, 1), YFLOWF( 1, 1), ZFLOWF( 1, 1),
     3    BOC    (156,3), AX          (3), AY           (3), AZ         (3)
       COMMON/CPRINT/      IPRG    , IPRH
       DIMENSION           CXO          (3),
     1    MTIP(3)       , MROOT(3)        , IVEL(4)
       CALL ERRSET (209,10,5,2)                                          DIV CHK
       CALL ERRSET (251,10,5,2)                                          NEG SQRT
       CALL ERRSET (252,10,5,2)                                          BIG EXPN
       CALL ERRSET (253,10,5,2)                                          NEG LG10
 7100  FORMAT( 16I5    )
 7200  FORMAT( 8F10.0 )
 7300  FORMAT( 80H
     1                                )
       READ(5,7300)
       READ(5,7200)        AMACH
       READ(5,7100)        NWP
       READ(5,7100)        (NCHORD(IW),      NSPAN(IW), IW=1,NWP)
       READ(5,7100)        (MTIP(IW),        MROOT(IW), IW=1,NWP)
       READ(5,7100)        NSYM
       READ(5,7100)        (IVEL(ISOLVE), ISOLVE=1,4)
       READ(5,7100)         IPRG   ,    IPRH
       READ(5,7200)        EPS    ,    EPSR
       DO 1100   IW=1,NWP
       AX(IW)        = - 1.
       AY(IW)        = 0.
 1100  AZ(IW)        = 0.
C
C
C
C
C      FOLLOWING FORMULAS FOR NETWORK POINTS APPLICABLE ONLY TO M-117 BOMB
       XNET( 5,1,1) = 4.345
       XNET( 9,1,1) = 3.435
       XNET(11,1,1) = 2.935
       XNET(15,1,1) = 2.035
       XNET(19,1,1) = 1.058
       XNET(24,1,1) = 0.
C
       DO 1254        I=1,8
 1254  XNET(I,1,1)   = XNET(5,1,1) +  ( I-5) * (XNET(11,1,1)-XNET(5,1,1))
     1                                / (11-5)
       XNET(10,1,1) = ( XNET(9,1,1) + XNET(11,1,1) )  /  2.
C
       DO 1256        I=12,14
 1256  XNET(I,1,1)   = XNET(11,1,1)+  ( I-11)* (XNET(15,1,1)-XNET(11,1,1))
     1                                / (15-11)
C
       DO 1258        I=16,18
 1258  XNET(I,1,1)   = XNET(15,1,1)+  ( I-15)* (XNET(19,1,1)-XNET(15,1,1))
     1                                / (19-15)
C
       DO 1260        I=20,23
 1260  XNET(I,1,1)   = XNET(19,1,1)+  ( I-19)* (XNET(24,1,1)-XNET(19,1,1))
     1                                / (24-19)
       NNC           = NCHORD(1)
       NNS           = NSPAN(1)
       DO 1262        I=1,NNC
       DO 1262        J=2,NNS
 1262  XNET(I,J,1)   = XNET(I,1,1)
C
       XO            = 1.055
       ZO            = - SQRT( 1.6**2 - XO**2 )
       ZNET( 1,1,1) = -.2815
       ZNET(11,1,1) = -.2815
       ZNET(15,1,1) = -.4
       ZNET(19,1,1) = -.4
       ZNET(24,1,1) = 0.
C
       DO 1294        I=1,NNC
       IF( I-11 )    1282, 1282, 1272
 1272  IF( I-15 )    1284, 1284, 1274
 1274  IF( I-19 )    1286, 1286, 1276
 1276  IF( I-24 )    1288, 1290, 1290
C
C      I=1,11
 1282  ZCL           = - ZNET(I,1,1)
       GO TO 1292
C
C      I=12,15
 1284  ZCL           = ZNET(11,1,1)+  ( I-11)* (ZNET(15,1,1)-ZNET(11,1,1))
     1                                / (15-11)
       ZCL           = - ZCL
       GO TO 1292
C
C      I=16,19
 1286  ZCL           = - ZNET(15,1,1)
       GO TO 1292
C
C      I=20,23
 1288  ZCL           =   ZO
     1                 + SQRT( 1.6**2 - (XNET(I,1,1)-XO)**2 )
       GO TO 1292
C
C      I=24
 1290  ZCL           = 0.
C
 1292  YNET(I,1,1)   = 0.
       YNET(I,2,1)   = ZCL * SIN( 3.1416 / 4. )
       YNET(I,3,1)   = ZCL
       YNET(I,4,1)   = ZCL * SIN( 3.1416 / 4. )
       YNET(I,5,1)   = 0.
       ZNET(I,1,1)   = - ZCL
       ZNET(I,2,1)   = - ZCL * SIN( 3.1416 / 4. )
       ZNET(I,3,1)   =   0.
       ZNET(I,4,1)   =   ZCL * SIN( 3.1416 / 4. )
       ZNET(I,5,1)   =   ZCL
 1294  CONTINUE
C
C
       IF( NWP-1 )   1380, 1380, 1300
 1300  NNC    = NCHORD(2)
       NNCM1  = NCHORD(2)-1
       NNS    = NSPAN(2)
       NNSM1  = NSPAN(2) -1
```

```
      XNET(1  ,1  ,2) =  3.9450
      XNET(1  ,NNS,2) =  3.4350
      XNET(NNC,1  ,2) =  4.3950
      XNET(NNC,NNS,2) =  4.3950
      DO 1312          I=1,NNC
      XNET(I  ,1  ,2) = XNET(1,1  ,2) + ( XNET(NNC,1,2(-XNET(1,1,2I )
     1                  * FLOAT(I-1) / (NNC-1I
      XNET(I  ,NNS,2) = XNET(1,NNS,2)
     1                 +( XNET(NNC,NNS,2)-XNET(1,NNS,2I I
     2                  * FLOAT(I-1) / (NNC-1I
         DO 1310       J=1,NNS
 1310    XNET(I,J,2)  = XNET(I,1,2)    + ( XNET(I,NNS,2I-XNET(I,1,2I I
     1                  * FLOAT(J-1) / (NNS-1I
 1312 CONT(NUE
C
      YNET(1,1  ,2)=  0.4020
      YNET(1,NNS,2)=  0.1991
      ZNET(1,1  ,2)= -YNET(1,1  ,2)
      ZNET(1,NNS,2)= -YNET(1,NNS,2)
      DO 1350       J=1,NNS
      YNET(1,J,2)  =    YNET(1,1,2)  +  ( YNET(1,NNS,2I - YNET(1,1,2I (
     1                * FLOAT(J-1) / (NNS-1I
 1350 ZNET(1,J,2(  =    ZNET(1,1,2)  +  ( ZNET(1,NNS,2) - ZNET(1,1,7I (
     1                * FLOAT(J-1) / (NNS-1I
C
      00 1360       I=2,NNC
      00 1360       J=1,NNS
      YNET(I,J,2)  = YNET(1,J,2)
 1360 ZNET(I,J,2)  = ZNET(1,J,2)
C
      00 1370       I=1,NNC
      00 1370       J=1,NNS
      XNET(I,J,3)  =  X4ET(I,J,2)
      YNET(I,J,3)  =  YNET(I,J,2)
 1370 ZNET(I,J,3)  = -ZNET(I,J,2(
 1380 CONT(NUE
C
      DO 1420      IW=1,NWP
      NNC            = NCHORD(IW)
      NNS            = NSPAN(IW)
      DO 1420       I=1,NNC
      00 1420       J=1,NNS
      XNET(I,J,IW) = - XNET(I,J,IW)
 1420 YNET(I,J,IW) = - YNET((,J,IW)
C     ARОVE FORMULAS FOR NETWORK POINTS APPLICABLE ONLY TO M-I17 BOMB
C
C
C
      BETA           = SQRT( 1. - AMACH**2 I
      00 1500      IW=1,NWP
      NNC            = NCHORO(IW)
      NNS            = NSPAN(IW)
      00 1500       1=1,NNC
      00 1500       J=1,NNS
      YNET(I,J,IW( = BETA * YNET(I,J,IW)
 1500 ZNET(I,J,IW) = BETA * ZNET(I,J,IW)
C
C
      WRITE(6,7300)
      WRITE(6,8004I                     AMACH
 8004 FORMAT(/        '                 AMACH                    '/6F20.2I
      WRITE(6,8110)                      NWP
 8110 FORMAT(/        '                  NWP                     '/6I20  I
C
      WRITE(6,8120I                (NCHORD(IW), NSPAN(IW),  IW=I,NWP)
 8I20 FORMAT(/        '                  NCHORO(1I          NSPAN(1)
     1        NCHORO(2)             NSPAN(2)         NCHORO(3(
     2NSPAN(3)   '                                           /6I20  )
      WRITE(6,8040I                     (MTIP(IW(,           MROOT(IW),
     1                                    IW=1,NWP)
 8040 FORMAT(/        '                   MTIP(1I            MROOT(I1
     1          MT)P(2)            MROOT(2)           MTIP(3)
     2MROOT(3)  '                                            /6I20  )
C
      WRITE(6,8130)                       NSYM
 8130 FORMAT(/        '                   NSYM                   '/6I20  I
C
      WRITE(6,8090)                     ((IVEL(ISOLVE], ISOLVE=1,4(
 8090 FORMAT(/        '                   IVEL(1)             IVEL(2)
     I           IVEL(3)'             IVEL(4)                    '/6I20  )
C
C
      WRITE(6,8160I                       IPRG,                IPRH
 8160 FORMAT(/        '                   IPRG                 IPRH'/6I20  (
C
      WRITE(6,8170)                        EPS,                 EPSR
 8170 FORMAT(/        '                    EPS                  EPSR'/6E20.2)
      WR(TE(6,8140)   (IW, AX(IW), AY(IW), AZ(IW(,  IW=1,NWP)
 8140 FORMAT(///      ' TRAILING VORTEX DIRECTION COSINES '
     1  /  6X,    ' AX(IW)     AY(IW)     AZ(IWI '  / (1H 'IW=',I1,3F10.6I(
C
      DO 2130   IW=1,NWP
      NUC(IW,1I   = NCHORO(1W(-1
      NUS((W,1)   = NSPAN(IW) -1
      NUC(IW,2)   = NCHORO(1W)
      NUS((W,7)   = NSPAN(IW) -1
      NUC((W,3)   = NCHORD(IW)-1
 2130 NUS(IW,3(   = NSPAN(IW)
C
      NVOR( 1)              =                 (NCHORO( 1)-1I * (NSPAN( 1)-1)
      DO 2I40     ISOLVE=1,3
 2140 NVEL( 1,(SOLVE] =    NUC( 1,ISOLVE) * NUS( 1,ISOLVE)
      IF( NWP-1 )  215A, 215A, 214A
 2148 DO 2150          IW=2,NWP
      NVOR(IW)              =  NVOR(IW-1(  +  (NCHORO(IWI-1) * (NSPAN((WI-1(
      DO 2150     ISOLVE=1,3
 2150 NVEL(IW,ISOLVE) =    NVEL(IW-1,ISOLVE)
     1                +    NUC(IW,ISOLVE) * NUS(IW,ISOLVEI
 215A DO 2I60     ISOLVE=1,3
 2160 NUT(ISOLVE)           =  NVEL( NWP,ISOLVE)
      NVORT                 =  NVOR(NWP)
C
C     IF AKSI, ETA, ZETA ARE LOCATED DIFFERENTLY WITH RESPECT TO XNET,
C     YNET, ZNET THAN IS USED HEREIN, CORRESPONOING CHANGES MUST BE MADE
C     IN CSUBP CALCULATIONS (SUBROUTINE ACOEFF, ISOLVE=1(
      DO 2650   IW=1,NWP
      NVORC         = NCHORD(IW)-1
      NVORS         = NSPAN(IW) -1
      DO 2650   IP=1,NVORC
      I             = IP
      DO 2650   IO=1,NVORS
      J              = IO
      IF( IW-1 )  2610, 2610, 2620
 2610 IT             =                 (IP-1)*NVORS + IO
      GO TO 2630
 2620 IT             = NVOR(IW-1) + ((P-I)*NVORS + IO
 2630 CONTINUE
      AKSI(IP,IO,IW) = (  XNET(I   ,J,IW) + XNET(I   ,J+1,IW)
     1                  + XNET(I+1,J,IW)  + XNET(I+1,J+1,IW)   ) / 4.
      ETA (IP,IO,IW) = (  YNET(I   ,J,IW) + YNET(I   ,J+1,IW)
     1                  + YNET(I+1,J,IW)  + YNET(I+1,J+1,IW)   ) / 4.
      ZETA(IP,IO,IW) = (  ZNET(I   ,J,IW) + ZNET(I   ,J+I,IW)
     1                  + ZNET(I+1,J,IW)  + ZNET(I+1,J+1,IW)   ) / 4.
      CX             = XNET(I,J+I,IW) - XNET(I+1,J,IW)
      CY             = YNET(I,J+1,IW) - YNET(I+1,J,IW)
      CZ             = ZNET(I,J+1,IW) - ZNET(I+1,J,1W)
      DX             = XNET(I,J   ,IW) - XNET(I+1,J+1,IW)
      DY             = YNET(I,J   ,IW) - YNET(I+1,J+1,IW)
      DZ             = ZNET(I,J   ,IW) - ZNET(I+1,J+1,IW)
      CXD(1)         =    CY * DZ - DY * CZ
      CXD(2)         = - CX * DZ + DX * CZ
      CXO(3)         =    CX * DY - DX * CY
      ARSCXD         = SQRT( CXO(1)**2 + CXD(2)**2 + CXO(3(**2 )
      DO 2650    N=1,3
 2650 BDC(IT,N)   = CXD(N) / ARSCXO
C
      DO 8290   IW=1,NWP
      NNC            = NCHORO(IW)
      NNS            = NSPAN(IW)
      GO TO( 8210, 8220, 8230), IW
 8210 WRITE(6,8212)   IW
 8212 FORMAT(//  ' NETWORK POINTS ON BODY               IW=',I1,
     1    '          J=1 IS TOP (-Z) CL         J=NJ IS BOTTOM (+Z) CL ')
      GO TO 8250
 8220 WRITE(6,8222)   IW
 8222 FORMAT(//  ' NETWORK POINTS ON UPPER FIN    IW=',I1,
     1    '          J=1 IS T(P                 J=NJ IS ROOT             '(
      GO TO 8250
 8230 WRITE(6,8232)   IW
 8232 FORMAT(//  ' NETWORK POINTS ON LOWER FIN    IW=',I1,
     1    '          J=1 IS TIP                 J=NJ IS ROOT             ')
 8250 WRITE(6,8260)
 8260 FORMAT(/     ' XNET(I,J,IW)     ')
      DO 8262    I=1,NNC
 8262 WRITE(6,8264)  I, (XNET(I,J,IW), J=1,NNS)
 8264 FORMAT(       ' I=',I2,   9F10.4      )
      WRITE(6,8270)
 8270 FORMAT(/      ' YNET(I,J,IW)       ')
      DO 8272    I=1,NNC
 8272 WRITE(6,8274)  I, (YNET(I,J,IW), J=1,NNS)
 8274 FORMAT(       ' I=',I2,   9F10.4      )
      WRITE(6,8280)
 8280 FORMAT(/      ' 2NET(I,J,IW)       ')
      DO 8282    I=1,NNC
 8282 WRITE(6,8284)  I, (ZNET(I,J,IW), J=1,NNS)
 8284 FORMAT(       ' I=',I2,   9F10.4      )
 8290 CONTINUE
C
C
      DO 8388   IW=1,NWP
      NVORC          = NCHORO(IW)-1
      NVORS          = NSPAN(IW) -1
      WRITE(6,8310)  IW
 8310 FORMAT(//     ' BOUNDARY POINTS                  IW=',I1          )
      WRITE(6,8360)
 8360 FORMAT(/      ' AKSI(IP,IO,IW) '   )
      DO 8362   IP=1,NVORC
 8362 WRITE(6,8364)  IP,  (AKSI(IP,IO,IW), IO=I,NVORS)
 8364 FORMAT(       ' IP=',I2, 8F10.5      )
      WRITE(6,8370)
 8370 FORMAT(/      '  ETA(IP,IO,IW) '   )
      DO 8372   IP=1,NVORC
 8372 WRITF(6,8374)  IP, ( ETA(IP,IO,IW), IO=1,NVORS)
 8374 FORMAT(       ' IP=',I2, 8F10.5      )
      WRITE(6,8380)
 8380 FORMAT(/      ' ZETA(IP,IO,IW) '   )
      DO 8382   IP=1,NVORC
 8382 WRITE(6,8384)  IP,  (ZETA(IP,(O,IW), IO=1,NVORS)
 8384 FORMAT(       ' IP=',I2, 8F10.5      )
 8388 CONTINUE
      WRITE(6,8410)
 8410 FORMAT(//     ' BDC(IT,N)    DIRECTION COSINES AT BOUNDARY POINTS
     1      N=1 IS X-COMPONENT        N=2 IS Y       N=3 IS Z    ')
      DO 8420       IW=1,NWP
      NVORC             = NCHORD(IW)-1
      NVORS             = NSPAN(IW) -1
      DO 8420        N=1,3
      DO 8420       IP=1,NVORC
      IF( IW-1 )  8412, 8412, 8414
 8412 IT1               =               (IP-1)*NVORS + 1
      IT2               =                IP   *NVORS
      GO TO 8416
 8414 IT1               = NVOR(IW-I) + (IP-1)*NVORS + 1
      IT2               = NVOR(IW-1) +  IP   *NVORS
 8416 WRITE(6,8418)  IW, N, IP,    (BDC(IT,N), IT=IT1,IT2)
 8418 FORMAT(       ' IW=',I1, '     N=',I1, '      IP=',I2,  8F10.5 )
 8420 CONTINUE
C
C
      NIWP          = NWP
      DO 3400   ISOLVE=1,4
      IF( IVEL(ISOLVE) )  3400, 3400, 3310
 3310 IF( ISOLVE-4      )  3380, 3320, 3320
 3320 CONTINUE
C     THE FOLLOWING STATEMENTS, THROUGH 3372, ARE USED TO COMPUTE VELOCITY
C     AT POINTS OFF THE SURFACE OF THE AERODYNAMIC PLANFORM
      READ(5,7100)        NIWP
      DO 3350   IW=1,NIWP
      READ(5,7100)        NUC(IW,4), NUS(IW,4)
      NVELC         = NUC(IW,4)
      NVELS         = NUS(IW,4)
      DO 7710   II=1,NVELC
 7710 READ(5,7200)        (XFLOWF(II,JJ), JJ=1,NVELS)
      DO 7720   II=1,NVELC
 7720 READ(5,7200)        (YFLOWF(I(,JJ), JJ=1,NVELS)
      DO 7730   II=1,NVELC
 7730 READ(5,7200)        (ZFLOWF(I(,JJ), JJ=1,NVELS)
 3350 CONTINUE
C
 3360 NVEL(1,4)   = NUC(1,4) * NUS(1,4)
```

```
      IF( NIWP-1 )  3372, 3372, 3364
 3364 DO 3370  IW=2,NIWP
 3370 NVEL(IW,4) = NVEL(IW-1,4)  +  NUC(IW,4) * NUS(IW,4)
 3372 NUT(4)     = NVEL(NIWP,4)
C
 3380 NVELT      = NUT(ISOLVE)
      CALL       ICOEFF( NVELT,   NVORT ,   ISOLVE,   NSYM  ,
     1   MTIP          , MROOT          ,
     1   EPS           , EPSR           )
 3400 CONTINUE
C
      IF( IVEL(1) )  4200, 4200, 4100
 4100 END FILE 23
      REWIND 23
 4200 END FILE 24
      REWIND 24
      STOP
      END
      SUBROUTINE ICOEFF( NVELT,    NVORT ,    ISOLVE,    NSYM  ,
     1    MTIP            , MROOT             ,
     1    EPS             , EPSR              )
      COMMON/CNET  /        NWP               , NIWP              ,
     1    NCHORD(3)       , NSPAN (3)         , NSCORD(3)         , NVOR  (3)     ,
     2    NUC    (3,4)    , NUS    (3,4)      , NVEL   (3,4)      , NUT    (4)
      COMMON/CCOORD/        XNET (24,5,3), YNET (24,5,3), ZNET (24,5,3),
     1                      AKSI (23,4,3), ETA  (23,4,3), ZETA (23,4,3),
     2                      XFLOWF( 1, 1), YFLOWF( 1, 1), ZFLOWF( 1, 1),
     3    BOC    (156,3), AX           (3), AY           (3), AZ           (3)
-     COMMON/CPRINT/        IPRG     , IPRM
      DIMENSION
     1    MTIP(3)         , MROOT(3)         ,
     2                                         H         (156), G      (156,3),
     2    R(2)            , COSTH(2)         , ASEG         (3), OSEG         (3),
     3    DIC    (24,3), EIC   (24,2,3), FIC        (2,3), SUMEIC  (2,3)
      APOKLF(A,B,C,D,E,F)  =  (A-B)*(C-E) - (D-E)*(F-B)
      NJSYM    = 3-NSYM
C
C
      DO 6200  LSYM=1,NSYM
      DO 6000     IW=1,NIWP
      NVELC           = NUC(IW,ISOLVE)
      NVELS           = NUS(IW,ISOLVE)
      DO 6000     II=1,NVELC
      DO 6000     JJ=1,NVELS
      IF( IW-1 )  510, 510, 520
  510 IT              =                      (II-1)*NVELS + JJ
      GO TO 530
  520 IT              = NVEL(IW-1,ISOLVE) + (II-1)*NVELS + JJ
  530 GO TO( 610, 620, 630, 640), ISOLVE
  610 IP              = II
      IQ              = JJ
      XP              = AKSI(IP,IQ,IW)
      YP              =  ETA(IP,IQ,IW)
      ZP              = ZETA(IP,IQ,IW)
      GO TO 700
C
  620 I               = II
      J               = JJ
      XP              = ( XNET(I,J,IW) + XNET(I,J+1,IW) ) / 2.
      YP              = ( YNET(I,J,IW) + YNET(I,J+1,IW) ) / 2.
      YP              = - ABS( YP )
      ZP              = ( ZNET(I,J,IW) + ZNET(I,J+1,IW) ) / 2.
      GO TO 700
C
  630 I               = II
      J               = JJ
      XP              = ( XNET(I,J,IW) + XNET(I+1,J,IW) ) / 2.
      YP              = ( YNET(I,J,IW) + YNET(I+1,J,IW) ) / 2.
      YP              = - ABS( YP )
      ZP              = ( ZNET(I,J,IW) + ZNET(I+1,J,IW) ) / 2.
      GO TO 700
C
  640 XP              = XFLOWF(II,JJ)
      YP              = YFLOWF(II,JJ)
      ZP              = ZFLOWF(II,JJ)
C
  700     DO 5200 JSYM=1,NJSYM
          DO 5000 JW=1,NWP
          NNC          = NCHORD(JW)
          NVORC        = NCHORD(JW)-1
          NVORS        = NSPAN(JW) -1
          DO 5000  L=1,NVORS
              DO 1500  K=1,NVORC
              IF( JW-1 )  1110, 1110, 1120
 1110         JT          =                 (K-1)*NVORS + L
              GO TO 1150
 1120         JT          = NVOR(JW-1) + (K-1)*NVORS + L
 1150         CONTINUE
C
              IF( ISOLVE-2 )  1250, 1220, 1250
 1220         IF( LSYM-JSYM)  1250, 1230, 1250
C1220         IS EQUIVALENT TO  LSYM=I  AND  JSYM=1
 1230         IF( IW-JW     )  1250, 1232, 1250
 1232         IF( II-K      )  1250, 1234, 1250
 1234         IF( JJ-L      )  1250, 1240, 1250
 1240         CONTINUE
              WRITE(6,8100)  LSYM,IW,II,JJ,  JSYM,JW,K,L
 8100         FORMAT(/  ' ICOEFF 8100 COUNTER TEST DELETES SPAN SEGMENT'
     1               ,4I5, 10X, 4I5   )
              GO TO 1450
 1250         CONTINUE
C
              EL    = SQRT(   ( XNET(K,L+1,JW) - XNET(K,L,JW)         )**2
     1                      + ( YNET(K,L+1,JW) - YNET(K,L,JW)         )**2
     2                      + ( ZNET(K,L+1,JW) - ZNET(K,L,JW)         )**2   )
              DO 1320 NN=1,2
 1320         R(NN)= SQRT(    ( XP                  - XNET(K,L-1+NN,JW) )**2
     1                      + ( YP                  - YNET(K,L-1+NN,JW) )**2
     2                      + ( ZP                  - ZNET(K,L-1+NN,JW) )**2   )
C
              DO 1334 NN=1,2
              IF( EL*R(NN) - EPS   )  1332, 1332, 1330
 1330         COSTH(NN) =    (R(1)**2 - R(2)**2 - (-1)**(NN) * EL**2 )
     1                     / (2.*EL*R(NN))
              GO TO 1334
 1332         COSTH(NN) = 0.D0
              WRITE(6,8112)  LSYM,IW,II,JJ, JSYM,JW,K,L, EL,NN,R(NN)
 8112         FORMAT(    ' ICOEFF 8112  EL*R(NN)< EPS     ON SPAN SEGMENT'
     1               ,4I5, 10X, 4I5, 10X, E10.2, I5, E10.2 )
 1334         CONTINUE
C
              IF( 1.-COSTH(2)**2 )  8130, 8130, 1340
 8130         WRITE(6,8132)  LSYM,IW,II,JJ, JSYM,JW,K,L, COSTH(2)
 8132         FORMAT(    ' ICOEFF 8132  COSTH(2)**2 > 1  ON SPAN SEGMENT'
     1               ,4I5, 10X, 4I5, 10X, F10.6      )
              GO TO 1450
C
 1340         SMALLR     = R(2) * SQRT( 1.-COSTH(2)**2 )
              IF( SMALLR - EPSR )  8150, 8150, 1350
 8150         WRITE(6,8152)  LSYM,IW,II,JJ,  JSYM,JW,K,L,  SMALLR
 8152         FORMAT(    ' ICOEFF 8152   SMALLR < EPSR    ON SPAN SEGMENT'
     1               ,4I5, 10X, 4I5, 10X, E10.2 )
              GO TO 1450
C
 1350         ASEG(1)=APOKLF( YNET(K,L+1,JW), YNET(K,L,JW), ZP,
     1                        ZNET(K,L+1,JW), ZNET(K,L,JW), YP )
              ASEG(2)=APOKLF( ZNET(K,L+1,JW), ZNET(K,L,JW), XP,
     1                        XNET(K,L+1,JW), XNET(K,L,JW), ZP )
```

```
            ASEG(3)=APOKLF( XNET(K,L+1,JWI, XNFT(K,L,JWI, YP,
     1                      YNET(K,L+1,JWI, YNET(K,L,JWI, XP )
            ABSLXR     = SQRT( ASEG(I)**2 + ASEG(2I**2 + ASEG(3)**2 I
            IF( ABSLXR - EPS )  8170, 8170, 1360
  8170      WRITE(6,8172)  LSYM,(W,II,JJ, JSYM,JW,K,L, ABSLXR
  8172      FORMAT(     ' ICOEFF 8172   ABSLXR < EPS    ON SPAN SEGMENT'
     1            ,4I5, 10X, 4I5, 10X, E10.2  )
            GO TO 1450
  1360      COEFF      =   `( COSTH(1) - COSTH(2) I
     1                   / ( 4.*3.1416 * SMALLR * ABSLXR )
            DO 1420  N=1,3
  1420      OIC(K,N)   = ASEG(N) * COEFF
            GO TO 1500
  1450      DO 1460  N=1,3
  1460      OIC(K,N)   = 0.DO
  1500      CONTINUE
C
C
            DO 2500 LR=1,2
            L1         = L-1+LR
            DO 2500  I=1,NVORC
            IF( L-1 )  2050, 2050, 2020
  2020      GO TO( 2030, 2050), LR
  2030.     DO 2040  N=1,3
  2040      EIC(I,1,N)= EIC(I,2,N)
            GO TO 2500
  2050      CONTINUE
C
C           ELIMINATION OF CHORDWISE SEGMENTS
            IF( 1SOLVE-3 )  2250, 2100, 2250
  2100      IF( LSYM-JSYM)  2210, 2110, 2210
C
  2110      LSYM = 1
            JSYM = 1
            IF( IW-JW I  2118, 2114, 2118
  2114      IF( I(-I  )  2250, 2116, 2250
  2116      IF( JJ-L1 )  2250, 2240, 2250
  2118      CONTINUE
            GO TO 2250
C
  2210      CONTINUE
            IF( IW-JW )       2250, 2222, 2250
  2222      IF( MTIP(IW) 1    2224, 2224, 2232
  2224      IF( II-I  )       2250, 2226, 2250
  2226      IF( JJ-L1 I       2250, 2228, 2250
  2228      IF( JJ-1  )       7000, 2240, 2232
C
  2232      IF( MROOT(IW) )  2234, 2234, 2250
  2234      IF( II-I  I       2250, 2236, 2250
  2236      IF( JJ-L1 )       2250, 2238, 2250
  2238      IF( JJ-NVELS I    2250, 2240, 2250
C
  2240      CONTINUE
            WRITE(6,8200)  LSYM,IW,II,JJ,  JSYM,JW,I,L1
  8200      FORMAT(/  ' ICOEFF 8200 COUNTER TEST DELETE CHORD SEGMENT'
     1            ,4I5, 10X, 4I5    )
            GO TO 2450
  2250      CONTINUE
C
C
            EL    = SQRT( ( XNET(I+1,L1,JW) - XNET(I       ,L1,JW) I**2
     1                  +( YNET(I+1,L1,JW) - YNET(I       ,L1,JW) )**2
     2                  +( ZNET(I+1,LI,JWI - ZNET(I       ,L1,JWI )**2 )
            DO 2320 NN=1,2
  2320      R(NN)= SQRT(  ( XP                - XNET(I-1+NN,L1,JWI I**2
     1                  +( YP                - YNET(I-1+NN,L1,JW) )**2
     2                  +( ZP                - ZNET(I-1+NN,L1,JWI I**2 I
            DO 2334 NN=1,2
            IF( EL*R(NN) - EPS  )  2332, 2332, 2330
  2330      COSTH(NN) =   (R(1)**2 - R(2)**2 - (-1I**(NNI * EL**21
     1                 / (2.*EL*R(NN))
            GO TO 2334
  2332      COSTH(NN) = 0.00
            WR(TE(6,8212)  LSYM,IW,I(,JJ, JSYM,JW,I,L1, EL,NN,R(NN)
  8212      FORMAT(     ' ICOEFF 8212  EL*R(NN)< EPS   ON CHORD SEGMENT'
     1            ,  4I5, 10X, 4I5, 10X, E10.2, I5, E10.2  1
  2334      CONTINUE
C
            IF( 1.-COSTH(2)**2 )  8230, 8230, 2340
  8230      WRITE(6,8232)  LSYM,IW,II,JJ, JSYM,JW,I,L1, COSTH(2)
  8232      FORMAT(     ' ICOEFF 8232  COSTH(2)**2 > 1 ON CHORD SEGMENT'
     1            ,  4I5, 10X, 4I5, 10X, F10.6    )
            GO TO 2450
C
  2340      SMALLR     = R(2) * SQRT( 1.-COSTH(2)**2 )
            IF( SMALLR - EPSR )  8250, 8250, 2350
  8250      WRITE(6,8252)  LSYM,IW,II,JJ, JSYM,JW,I,L1, SMALLR
  8252      FORMAT(     ' ICOEFF 8252   SMALLR < EPSR  ON CHORD SEGMENT'
     1            ,  4I5, 10X, 4I5, IOX, E10.2  )
            GO TO 2450
  2350      ASEG(1)    = APOKLF( YNET(I+1,L1,JW), YNET(I,L1,JW), ZP,
     1                           ZNET(I+1,L1,JW), ZNET(I,L1,JWI, YP )
            ASEG(2)    = APOKLF( ZNET(I+1,L1,JW), ZNET(I,L1,JW), XP,
     1                           XNET(I+1,L1,JW), XNET(I,L1,JW), ZP )
            ASEG(3)    = APOKLF( XNET(I+1,L1,JHI, XNET(I,L1,JWI, YP,
     1                           YNET(I+1,L1,JWI, YNET(I,L1,JW), XP )
            ABSLXR     = SQRT( ASEG(1)**2 + ASEG(2I**2 + ASEG(3)**2 I
            IF( ABSLXR - EPS )  8270, 8270, 2360
  8270      WRITE(6,8272)  LSYM,IW,II,JJ, JSYM,JW,I,L1, ABSLXR
  8272      FORMAT(     ' 1COEFF 8272   ABSLXR < EPS    ON CHORD SEGMENT'
     1            ,4I5, 10X, 4I5, 10X, E10.2  I
            GO TO 2450
  2360      COEFF      =    ( COSTH(1) - COSTH(2I )
     1                   / ( 4.*3.1416 * SMALLR * ABSLXR )
            DO 2420      N=1,3
  2420      EIC(I,LR,N) = ASEG(N) * COEFF
            GO TO 2500
  2450      DO 2460      N=1,3
  2460      E(C(I,LR,N) = 0.DO
  2500      CONTINUE
C
C
            DO 3500 LR=1,2
            L1         = L-1+LR
            R(I)       = SQRT(          ( XP  - XNET(NNC,L1,JWI )**2
     1                         +        ( YP  - YNET(NNC,L1,JW) )**2
     2                         +        ( ZP  - ZNET(NNC,L1,JWI )**2   )
            IF( R(1)-EPSR )  3332, 3332, 3330
  3330      COSTH(1)   = -(  AX(JW) * ( XP  - XNET(NNC,L1,JW) )
     1                     + AY(JW) * ( YP  - YNET(NNC,L1,JW) )
     2                    + AZ(JW) * ( ZP   - ZNFT(NNC,L1,JW1 1 1 /R(1)
            GO TO 3336
 3332       COSTH(1)   = 0.00
            WRITE(6,8312)  LSYM,IW,II,JJ, JSYM,JW,L1, R(1)
 8312       FORMAT(    ' ICOEFF 8312      R(1) < FPSR    ON TRAIL VORTFX'
     1           ,  415, ' TRAIL VOR',  215, 5X, 15, 10X,  E10.?  )
C
 3336       IF( 1.-COSTH(1)**2 )  R330, R330, 3340
 8330       WRITE(6,8332)  LSYM,IW,II,JJ, JSYM,JW,LI, COSTH(1)
 8332       FORMAT(    ' ICOEFF 8332  COSTH(I)**2 > I   ON TRAIL VORTEX'
     1           ,  415, ' TRAIL VOR',  215, 5X, 15, 10X,  F10.6  )
            GO TO 3450
 3340       SMALLR     = R(1) * SORT( 1.-COSTH(I(**2 (
            IF( SMALLR-EPSR )  8350, 8350, 3350
 8350       WRITE(6,8352)  LSYM,IW,II,JJ, JSYM,JW,LI, SMALLR
 8352       FORMAT(    ' ICOEFF 8352   SMALLR < FPSR    ON TRAIL VORTEX'
     1           ,  415, ' TRAIL VOR',  215, 5X, I5, IOX,  F10.2  1
            GO TO 3450
 3350       DSEG(1)    =      AY(JW) * ( ZP   - ZNET(NNC,L1,JW) (
     1                      - AZ(JW) * ( YP   - YNET(NNC,L1,JW) )
C
            DSEG(2)    =      AZ(JW) * ( XP   - XNFT(NNC,L1,JW1 (
     1                      - AX(JW) * ( ZP   - ZNET(NNC,L1,JW1 )
C
            OSEG(3)    =      AX(JW) * ( YP   - YNET(NNC,L1,JW1 1
     1                      - AY(JW) * ( XP   - XNFT(NNC,L1,JW( (
            ABSLXR     = SORT( OSFG(1)**2 + DSEG(2(**2 + DSEG(3(**2 (
            IF( ABSLXR - EPS )  8370, 8370, 3360
 8370       WRITE(6,8372)  LSYM,IW,I1,JJ, JSYM,JW,LI, ABSLXR
 8372       FORMAT(    ' ICOEFF 8372   ABSLXR < EPS     ON TRAIL VORTEX'
     1           ,  415, ' TRAIL VOR',   215, 5X,  15, 10X, E10.2 1
            GO TO 3450
 3360       COEFF      =    ( 1. - COSTH(1) )
     1                    / ( 4.*3.1416 * SMALLR * ABSLXR 1
            00 3420  N=1,3
 3420       FIC(LR,N( = OSEG(N) * COEFF
            GO TO 3500
 3450       DO 3460  N=1,3
 3460       FIC(LR,N) = 0.00
 3500       CONTINUE
C
C
            00 4100      N=1,3
            00 4100    LR=1,2
 4100       SUMEIC(LR,N) = 0.00
C
            00 4300      M=1,NVORC
            K = NCHORO(JW)-M
            IF( JW-1 1  4210, 4210, 4220
 4210       JT               =                (K-1(*NVORS + L
            GO TO 4230
 4220       JT               = NVOR(JW-I) + (K-1(*NVORS + L
 4230       00 4300      N=I,3
                00 4240    LR=1,2
 4240           SUMEIC(LR,N) = SUMEIC(LR,N) + EIC(K,LR,N(
                GIC          = OIC(K,N) - SUMEIC(I,NI + SUMEIC(2,N)
     1                                  -       FIC(1,N( +     FIC(2,N)
                GO TO( 4250, 4252), NSYM
 4250           GO TO( 4252, 4254), JSYM
 4252           G(JT,N)      = GIC
                GO TO 4300
 4254           G(JT,N)      = G(JT,N) - GIC
 4300       CONTINUE
 5000   CONTINUE
C
        GO TO( 5090, 5200), NSYM
 5090   00 5100 JW=1,NWP
        NNC        = NCHORD(JW)
        NNS        = NSPAN(JW)
        00 5100  K=1,NNC
        00 5100  L=1,NNS
 5100   YNET(K,L,JW) = - YNET(K,L,JW)
C
 5200   CONTINUE
C
C
        IF( IPRG      )  8550, 8550, 8512
 8512   IF( IT-1      )  8516, 8516, 8514
 8514   IF( IT-NVELT )   8550, 8516, 8516
 8516   DO 8520  N=1,3
        WRITE(6,8518)  ISOLVE, LSYM, IT, N
 8518   FORMAT(        ' ISOLVE =',11,  '  LSYM=',11, '  IT=',13, '  N=',
     1     (1, 10X,  ' G(JT,N)  AS ORIGINALLY COMPUTEO '(
 8520   WRITE(6,8522) (G(JT,N), JT=1,NVORT(
 8522   FORMAT( 16F8.3 )
 8530   WRITE(6,8540)  IT
 8540   FORMAT(75X,'   ICOEFF    IT=',I3,   '    ENTRY WRITE(24) G '(
 8550   WRITE(24)      G
C
        GO TO( 5500, 6000, 6000, 6000), ISOLVE
 5500   00 5520 JT=1,NVORT
        H(JT)      = 0.00
        DO 5520  N=1,3
 5520   H(JT)      = H(JT)  +  8DC(IT,N) * G(JT,NI
        WRITE(23)      H
        IF( IPRH      )  6000, 6000, 8562
 8562   IF( IT-1      )  8570, 8570, 8564
 8564   IF( IT-NVORT )  6000, 8570, 8570
 8570   WRITE(6,8572)  LSYM, IT, (H(JT), JT=1,NVORT(
 8572   FORMAT(       ' H(JT)  NON-NORMALIZED AS ORIGINALLY COMPUTEO
     1            LSYM=',I1,   '  IT=',I3      /     (1H 16F8.3)     )
 6000 CONTINUE
C
      GO TO( 6200, 6090), NSYM
 6090 DO 6100      IW=1,NWP
      NNC             = NCHORO(IW)
      NNS             = NSPAN(IW)
      00 6100       1=1,NNC
      00 6100       J=1,NNS
 6100 YNET(I,J,IW) = - YNET(1,J,IW)
C
 6200 CONTINUE
C
 7000 RETURN
      ENO
```

Output Data

The output data from Program A consists of the $\vec{G}$ and H matrices. The $\vec{G}$ matrices are written on a magnetic tape identified with data set reference number 24, and the H matrix is written on a second magnetic tape having the data set reference number 23. The entire quantity of output data which is generated in the sample run that is used herein as a benchmark is much to voluminous to display in its entirety; hence, only representative results will be shown.

When NSYM = 2 and ISOLVE = 1, the elements of the first row and first column of the x, y, z components of $\vec{G}^{(1)}$ (the first partition of $\vec{G}$, the FORTRAN designation for the first partition being LSYM = 1) are

$$G^{(1)}_{x_{1,1}} = 0,\ G^{(1)}_{y_{1,1}} = 1.321,\ \text{and}\ G^{(1)}_{z_{1,1}} = 3.188$$

The corresponding value of H is

$$H^{(1)}_{1,1} = -3.451$$

The initial elements of the second partition (LSYM = 2) are

$$G^{(2)}_{x_{1,1}} = 0.126,\ G^{(2)}_{y_{1,1}} = 0.621,\ \text{and}\ G^{(2)}_{z_{1,1}} = 0.648;$$

$$\text{and}\ H^{(2)}_{1,1} = -0.836$$

2.2 USER'S GUIDE TO PROGRAM B

Input Data

Several of the input variables whose values are read from punched card data sets are mentioned above in the description of Program A. The others are defined as follows:

IPRHI = 0	Do not print H^{-1}
= 1	Print H^{-1}
ISEG = 1	Entire Program B is executed in one submittal to the computer
= 2	Program B is segmented into 2 parts, each of which requires a separate submittal

IRUN = 1 This is the first submittal
= 2 This is the second submittal when ISEG = 2; it is a continuation of calculations performed in the first submittal

The $\vec{G}$ and H matrices, which were written in Program A as data sets with reference numbers 24 and 23, respectively, are read in Program B.

Dimensions of Arrays

The values of the dimensions for the variables in COMMON/CNET/ are the same as used in Program A. Wherever the number 156 appears as a dimension in the sample program listing, it indicates that the dimensions in those locations are equal to NVORT.

PROGRAM B
INPUT DATA FROM PUNCHED CARDS
SAMPLE RUN

```
 H-INVERSE MATRIX FOR AN M-117 BOMB AT M=0.0 IS COMPUTED FOR SAMPLE RUN
    3

   24    5    9    5    9    5
    2

    1    1    1    0
    1    1    1

    1    1
```

```
C        PROGRAM 8
C        IN THIS PROGRAM THE H MATRIX IS INVERTED AND
C        THE HDIAG, HINV, H AND G MATRICES ARE WRITTEN ON DATA SET 21
C        1 SQUARE MATRIX IN-CORE
C        NON-PIVOTING
C        ISEG = 1        NON-SEGMENTED, IRUN=1
C        ISEG = 2        SEGMENTED INTO 2 RUNS, IRUN=1,2
C
C        ISEG = 1
C        WRITE(21)
C        READ(23)
C        READ(24)
C        WRITE(31)     IH1(IT,JT), JT=1,NVORT)    NORMALIZED
C        READ(31)       EH1(JT)
C        WRITE(32)      EH2(JT)                   NORMALIZED
C        READ(32)      IH2I(T,JT), JT=1,NVORT)
C        WRITE(41)      H2HI(JT)
C        WRITE(42)      EH1H2(JT)                 =                EH1 - H2HIH2
C        READ(42)      (HINV1(IT,JT), JT=1,NVORTI = INVERSE OF (EH1 - H2HIH2I
C        WRITE(51)     (H1INV(IT,JT), IT=1,NVORTI = INVERSE OF    H1
C        READ(51)       EH1INV(IT)
C        WRITE(52)      H2HI(IT)
C        WRITE(53)      EHINV2(IT)                = - HINV1 * H2HI
C        READ(53)      IH1NV2I(T,JT1, IT=1,NVORTI
C
C        ISEG = 2  AND  IRUN=1
C        READ(23)
C        WRITE(22)      HDIAG, EH1H2, H2HI(IT), H1, H2
C        UNIT(31)       H1
C            (32)       H2
C            (41)       H2HI(JT)
C            (51)       H1INV
C
C        ISEG = 2  AND  IRUN=2
C        WRITE(21)
C        READ(22)
C        READ(24)
C        UNIT(53)
         COMMON/CNET  /      NWP              , NIWP              ,
       1   NCHORD(3)      , NSPAN (3)       , NSCORD(3I        , NVOR  (3)
       2   NUC    (3,4)   , NUS   (3,4)     , NVEL   (3,4I     , NUT     (4I
         DIMENSION              IVEL(4)
         DIMENSION
       1                                                        G     (156,3)
         DIMENSION
       1   H1   (156,156),H1INV(156,156), H2  (156,1561,HINV1(156,1561,
       2  HINV2(156,156)
         DIMENSION
       1   HDIAG (156,2),
       2   EH2        (156), H2HI     (156), H2HIH2  (1561, EH1       (1561,
       3   EH1H2     (156),
       1   EHINV1    (156),
       1   EH1INV(156)    , EHINV2(156)
C
         EQUIVALENCE
       1I  H1   (   1,   1I, H1INV( 1,   1), H2   I   1,   1I, HINV1( 1,   1I,
       2   HINV2( 1,   1)   )
         CALL ERRSET (209,10,5,2)                                                DIV CHK
         CALL ERRSET (251,10,5,2)                                                NEG SORT
         CALL ERRSET (252,10,5,2)                                                BIG EXPN
         CALL ERRSET I253,10,5,2)                                                NEG LG10
  7100  FORMAT( 1615    )
  7200  FORMAT( 8E10.0 )
  7300  FORMAT( 80H
       1                                   )
         READ(5,7300)
         READ(5,7100)        NWP
         READ(5,7100)       (NCHORD(IW),     NSPAN(IW),   IW=1,NWP1
         READ(5,7100)        NSYM
         READ(5,7100)       I IVEL(ISOLVE), ISOLVE=1,4I
         READ(5,7100)        IPRHI ,    IPRH   ,    IPRG
         READ(5,7100)        ISEG   ;    IRUN
         DO 2130   IW=1,NWP
         NUC(IW,1)   = NCHORD(IW)-1
         NUS(IW,1)   = NSPAN(IW) -1
         NUC)IW,2)   = NCHORD(IW)
         NUS(IW,2)   = NSPAN(IW) -1
         NUC(IW,3)   = NCHORD(IW)-1
  2130   NUS(IW,3)   = NSPAN(IW)
C
         NVOR( 1)              =                     (NCHORD( 1)-1) * (NSPAN( 1)-1(
         DO 2140     ISOLVE=1,3
  2140   NVEL( 1,ISOLVE) =    NUC( 1,ISOLVE) * NUS( 1,ISOLVE1
         IF( NWP-1 )  2158, 2158, 2148
  2148   DO 2150         IW=2,NWP
         NVOR(IW)              =    NVOR(IW-1)  +   (NCHORD(IW)-1I * INSPAN(IW)-1)
         DO 2150     ISOLVE=1,3
  2150   NVEL(IW,ISOLVE) =    NVEL(IW-1,ISOLVE)
       1                  +   NUC(IW,ISOLVE) * NUS(IW,ISOLVE1
  2158   DO 2160     ISOLVE=1,3
  2160   NUT(ISOLVE)           =    NVEL( NWP,ISOLVEI
         NVORT                 =    NVOR(NWP)
C
         WRITE(6,7300)
         WRITE(6,8110)                             NWP
  8110   FORMAT(/              '                   NWP                      '/6120   1
C
         WRITE(6,8120)                       (NCHORD(IW1, NSPAN(IW1,   IW=1,NWP1
  8120   FORMAT(/              '                    NCHORD(1I                NSPAN(1I
       1        NCHORD(2)                  NSPAN(2)           NCHORD(3)
       2NSPAN(3) '                                                           /6I20   1
         WRITE(6,8130)                             NSYM
  8130   FORMAT(/              '                   NSYM                     '/6120   1
         WRITE(6,8140)                      (IVEL(ISOLVE1, ISOLVE=1,4]
  8140   FORMAT(/              '                   IVEL(1I                  IVEL(2I
       1            IVEL(3)                 IVEL(4)                          '/6I20   1
         WRITE(6,8160)                            IPRHI,                   IPRH,
       1            IPRG,              IPRHH1
  8160   FORMAT(/              '                  IPRHI                    IPRH
       1            IPRG               IPRHH1                                '/6I20   1
         WRITE(6,8170)                             ISEG,                    IRUN
  8170   FORMAT(/              '                   ISEG                     IRUN'/6120   1
         WRITE(6,8210)  NVORT
  8210   FORMAT(  '1     COMPUTED VALUE OF NVORT=',I3   1
C
         GO TO( 3400, 4600), IRUN
  3400   CONTINUE
         DO 3440      IT=1,NVORT
         READ(23)          EH1
         DO 3410      JT=1,NVORT
```

```
 3410 H1(IT,JT)     = EH1(JT)
      IF( IPRH      )  8530, 8530, 8516
 8516 IF( IT-1      )  8520, 8520, 8518
 8518 IF( IT-NVORT )  8530, 8520, 8520
 8520 WRITE(6,8522) IT,(H1(IT,JT), JT=1,NVORT)
 8522 FORMAT( ' H1(IT,JT)  NON-NORMALIZED FROM READ(23)    IT=',I3
     1       ' /        (1H 16F8.3)     )
 8530 CONTINUE
      HDIAG(IT,1)   = H1(IT,IT)
      HDIAG(IT,2)   = - HDIAG(IT,1)
      DO 3430      JT=1,NVORT
      H1(IT,JT)     = H1(IT,JT) / HDIAG(IT,1)
 3430 EH1(JT)       = H1(IT,JT)
 3440 WRITE(31)        EH1
      END FILE 31
      REWIND 31
C
      I21            = (2-ISEG)*21 + (ISEG-1)*22
      WRITE(I21)       HDIAG
C
      DO 8610      IT=1,NVORT
 8610 WRITE(6,8612) IT,(HDIAG(IT,ISYM), ISYM=1,NSYM)
 8612 FORMAT(         '     HDIAG(IT,ISYM)       IT=',I3,   2E15.7 )
      IF( IPRH      )  8626, 8626, 8614
 8614 DO 8624      IT=1,NVORT
      IF( IT-1      )  8620, 8620, 8618
 8618 IF( IT-NVORT )  8624, 8620, 8620
 8620 WRITE(6,8622) IT,(H1(IT,JT), JT=1,NVORT)
 8622 FORMAT( ' H1(IT,JT)    NORMALIZED AS ORIGINALLY COMPUTED    IT=',I3
     1        /         (1H 16F8.3)   )
 8624 CONTINUE
 8626 CONTINUE
C
      CALL          MATINV( NVORT, H1INV )
C
      GO TO( 4100, 4200), NSYM
 4100 DO 4120      IT=1,NVORT
      DO 4110      JT=1,NVORT
 4110 EH1INV(JT)    = H1INV(IT,JT)
 4120 WRITE(21)        EH1INV
      DO 4130      IT=1,NVORT
      READ(31)         EH1
 4130 WRITE(21)        EH1
      REWIND 31
      GO TO 6400
C
 4200 CONTINUE
      DO 4250      JT=1,NVORT
      DO 4240      IT=1,NVORT
 4240 EH1INV(IT)    = H1INV(IT,JT)
 4250 WRITE(51)        EH1INV
      END FILE 51
      REWIND 51
C
      IF( IPRHI     )  8656, 8656, 8640
 8640 DO 8654      IT=1,NVORT
      IF( IT-1      )  8650, 8650, 8648
 8648 IF( IT-NVORT )  8654, 8650, 8650
 8650 WRITE(6,8652) IT,(H1INV(IT,JT), JT=1,NVORT)
 8652 FORMAT(          '      H1INV(IT,JT)              INVERSE OF H1        IT=',I3
     1        /      (1H  8E15.7)   )
 8654 CONTINUE
 8656 CONTINUE
C
      DO 4330      IT=1,NVORT
         READ(23)      EH2
         IF( IPRH      )  8580, 8580, 8566
 8566    IF( IT-1      )  8570, 8570, 8568
 8568    IF( IT-NVORT )  8580, 8570, 8570
 8570    WRITE(6,8572)  IT, (EH2(JT), JT=1,NVORT)
 8572    FORMAT(  ' EH2(JT) NON-NORMALIZED FROM READ(23)  IT=',I3
     1            / (1H 16F8.3)   )
 8580    CONTINUE
         DO 4310 JT=1,NVORT
 4310    EH2(JT)    = EH2(JT) / HDIAG(IT,1)
         WRITE(32)     EH2
         DO 4320 JT=1,NVORT
         H2HI(JT)   = 0.
         DO 4320 KK=1,NVORT
 4320    H2HI(JT)   = H2HI(JT)   +   EH2(KK) * H1INV(KK,JT)
         WRITE(41)     H2HI
 4330 CONTINUE
      END FILE 32
      END FILE 41
      REWIND 23
      REWIND 32
      REWIND 41
C
      DO 4350      IT=1,NVORT
      READ(32)         EH2
      DO 4350      JT=1,NVORT
 4350 H2(IT,JT)     = EH2(JT)
      REWIND 32
C
      IF( IPRH      )  8690, 8690, 8660
 8660 DO 8680      IT=1,NVORT
      IF( IT-1      )  8670, 8670, 8668
 8668 IF( IT-NVORT )  8680, 8670, 8670
 8670 WRITE(6,8672) IT,(H2(IT,JT), JT=1,NVORT)
 8672 FORMAT( ' H2(IT,JT)  NORMALIZED AS ORIGINALLY COMPUTED      IT=',I3
     1        /        (1H  8F15.7)   )
 8680 CONTINUE
 8690 CONTINUE
C
      I42            = (2-ISEG)*42 + (ISEG-1)*22
      DO 4430      IT=1,NVORT
         READ(41)      H2HI
         DO 4410 JT=1,NVORT
         H2HIH2(JT)= 0.
         DO 4410 KK=1,NVORT
 4410    H2HIH2(JT)= H2HIH2(JT)   +   H2HI(KK) * H2(KK,JT)
         READ(31)      EH1
         DO 4420 JT=1,NVORT
 4420    EH1H2(JT) = EH1(JT) - H2HIH2(JT)
         WRITE(I42)    EH1H2
 4430 CONTINUE
      REWIND 31
      REWIND 41
C
      I52            = (2-ISEG)*52 + (ISEG-1)*22
      DO 4470     JT=1,NVORT
         READ(51)      EH1INV
         DO 4460 IT=1,NVORT
         H2H1(IT)   = 0.
         DO 4460 KK=1,NVORT
 4460    H2H1(IT)   = H2H1(IT)   +  H2(IT,KK) * EH1INV(KK)
         WRITE(I52)  H2H1
 4470 CONTINUE
      REWIND 51
C
      GO TO( 4480, 4500), ISEG
 4480 END FILE 42
      END FILE 52
      REWIND 42
      REWIND 52
      GO TO 4700
C
 4500 DO 4510     IT=1,NVORT
      READ(31)         EH1
 4510 WRITE(22)        EH1
      DO 4520     IT=1,NVORT
      READ(31)         EH2
 4520 WRITE(22)        EH2
      END FILE 72
      REWIND 22
      STOP
C
C
 4600 READ(22)         HDIAG
      WRITE(21)        HDIAG
C
      I42              = 22
 4700 DO 4720     IT=1,NVORT
      READ(I42)        EHINV1
      DO 4720     JT=1,NVORT
 4720 HINV1(IT,JT) = EHINV1(JT)
C
      CALL        MATINV( NVORT, HINV1 )
C
      DO 4750     IT=1,NVORT
      DO 4740     JT=1,NVORT
 4740 EHINV1(JT)    = HINV1(IT,JT)
      WRITE(21)        EHINV1
      IF( IPRHI     )  4750, 4750, 8716
 8716 IF( IT-1      )  8720, 8720, 8718
 8718 IF( IT-NVORT  )  4750, 8720, 8720
 8720 WRITE(6,8722) IT,(HINV1(IT,JT), JT=1,NVORT)
 8722 FORMAT(       '     HINV1(IT,JT)          INVERSE OF H1-H2*H1INV*H2
     1          IT=',I3     /     (1H  8E15.7)   )
 4750 CONTINUE
C
      I52              = (2-ISEG)*52 + (ISEG-1)*22
      DO 4820     JT=1,NVORT
         READ(I52)   H2HI
         DO 4810 IT=1,NVORT
         EHINV2(IT)= 0.
         DO 4810 KK=1,NVORT
 4810    EHINV2(IT)= EHINV2(IT)   -  HINV1(IT,KK) * H2HI(KK)
         WRITE(53)   EHINV2
 4820 CONTINUE
      END FILE 53
      REWIND 53
C
      DO 4850     JT=1,NVORT
      READ(53)         EHINV2
      DO 4850     IT=1,NVORT
 4850 HINV2(IT,JT) = EHINV2(IT)
C
      DO 4860     IT=1,NVORT
      DO 4858     JT=1,NVORT
 4858 EHINV2(JT)    = HINV2(IT,JT)
      WRITE(21)        EHINV2
      IF( IPRHI     )  4860, 4860, 8766
 8766 IF( IT-1      )  8770, 8770, 8768
 8768 IF( IT-NVORT  )  4860, 8770, 8770
 8770 WRITE(6,8772) IT,(HINV2(IT,JT), JT=1,NVORT)
 8772 FORMAT(       '     HINV2(IT,JT) = - HINV1 * H2 * H1INV
     1          IT=',I3     /     (1H  8E15.7)   )
 4860 CONTINUE
C
      I31              = (2-ISEG)*31 + (ISEG-1)*22
      I32              = (2-ISEG)*32 + (ISEG-1)*22
      DO 4910     IT=1,NVORT
      READ(I31)        EH1
 4910 WRITE(21)        EH1
C
      DO 4922     IT=1,NVORT
      READ(I32)        EH2
      WRITE(21)        EH2
      GO TO( 4922, 4920), ISEG
 4920 REWIND 22
 4922 CONTINUE
C
C
 6400 CONTINUE
      DO 6480   ISOLVE=1,4
      IF( IVEL(ISOLVE) )  6480, 6480, 6426
 6426 NVELT            = NUT(ISOLVE)
      DO 6460   ISYM=1,NSYM
      DO 6460     IT=1,NVELT
      READ(24)         G
      WRITE(21)        G
      IF( IPRG      )  6460, 6460, 8912
 8912 IF( IT-1      )  8916, 8916, 8914
 8914 IF( IT-NVELT  )  6460, 8916, 8916
 8916 DO 8920        N=1,3
 8920 WRITE(6,8922)       ISOLVE, ISYM, IT, N,  (G(JT,N), JT=1,NVORT)
 8922 FORMAT(        ' ISOLVE=',I1,   5X, ' ISYM=',I1,   5X, ' IT=',I3,  5X,
     1       ' N=',I1, 15X, ' G(JT,N) READ FROM UNIT(24)' / (1H 16F8.3))
 6460 CONTINUE
 6480 CONTINUE
      END FILE 21
      REWIND 21
      REWIND 24
      STOP
      END
      SUBROUTINE MATINV( NVORT,    A )
      DIMENSION            INDEX (156,3), A(NVORT,NVORT)
      DOUBLE PRECISION
     2   AMAX            , SWAP            , PIVOT           , T
      NIM2 = NVORT
C
C
C     REFERENCE    MCCORMICK/SALVADORI  P.306,  KUO  P.168
C
C        INITIALIZATION
C        INDEX(J,3)  OF M/S  =  IPVOT(J)  OF KUO
         DO 52     J=1,NIM2
   52    INDEX(J,3)= 0
C
C
         DO 550           I=1,NIM2
         IROW         = I
         ICOLUM       = I
C        SINCE THE H-MATRIX IS DIAGONALLY DOMINANT BYPASS PIVOTING
         GO TO 140
C            SEARCH FOR PIVOT ELEMENT
             AMAX         = 0.D0
            DO 105      J=1,NIM2
             IF( INDEX(J,3) - 1 ) 60, 105, 60
   60            DO 100     K=1,NIM2
                 IF( INDEX(K,3) - 1 )           80, 100, 715
   80            IF( AMAX -  ABS(A(J,K)) )  85, 100, 100
   85            IROW         = J
                 ICOLUM       = K
                 AMAX         =  ABS( A(J,K) )
  100            CONTINUE
  105        CONTINUE
  140    CONTINUE
             INDEX(ICOLUM,3) = INDEX(ICOLUM,3) + 1
             INDEX(I      ,1) = IROW
             INDEX(I      ,2) = ICOLUM
C
C            INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL
C
             IF( IROW - ICOLUM ) 150, 310, 150
  150        DO 200     L=1,NIM2
             SWAP         = A(IROW   ,L)
             A(IROW   ,L)= A(ICOLUM,L)
  200        A(ICOLUM,L)= SWAP
C
C            DIVIDE PIVOT ROW BY PIVOT ELEMENT
C
  310        PIVOT         = A(ICOLUM,ICOLUM)
             A(ICOLUM,ICOLUM)= 1.D0
             DO 350     L=1,NIM2
  350        A(ICOLUM,L)= A(ICOLUM,L) / PIVOT
C
C            REDUCE NON-PIVOT ROWS
         DO 550           L1=1,NIM2
             IF( L1 - ICOLUM )  400, 550, 400
  400        T             = A(L1,ICOLUM)
             A(L1,ICOLUM)= 0.D0
             DO 450     L=1,NIM2
  450        A(L1,L)       = A(L1,L)  -  A(ICOLUM,L) * T
  550    CONTINUE
C
C        INTERCHANGE COLUMNS
C
         DO 710           I=1,NIM2
             L             =  NIM2 + 1 - I
             IF( INDEX(L,1) - INDEX(L,2) )  630, 710, 630
C
  630        JROW          = INDEX(L,1)
             JCOLUM        = INDEX(L,2)
C
             DO 705     K=1,NIM2
             SWAP          = A(K,JROW)
             A(K,JROW)  = A(K,JCOLUM)
             A(K,JCOLUM)= SWAP
  705        CONTINUE
  710    CONTINUE
C
C        TEST FOR SINGULARITY OF MATRIX
         DO 730    K=1,NIM2
             IF( INDEX(K,3) - I )  715, 730, 715
  715        WRITE(6,8100)
 8100        FORMAT('1  SINGULAR MATRIX OCCURED IN SUBROUTINE MATINV')
             STOP
  730    CONTINUE
C
      RETURN
      END
```

Output Data

Values of the HDIAG and normalized H matrices may be inferred from the output data of Program A. Representative values for the illustrative case described herein are shown for elements of these matrices in addition to the partitions of the H^{-1} matrix, $A^{(1)}$ and $A^{(2)}$.

First Partition		Second Partition	
HDIAG(1, 1)	= -3.45	HDIAG(1, 2) =	3.45
$H^{(1)}_{1,1}$	≡ 1.000	$H^{(2)}_{1,1}$	= 0.242
$A^{(1)}_{1,1}$	= 1.082	$A^{(2)}_{1,1}$	= -0.314

2.3 USER'S GUIDE TO PROGRAM C—FIRST JOB STEP

Input Data

The user must define the following variables by arithmetic expressions:

NSYM — This is defined in Section 2. 1 of this Appendix

NVORT — Number of vortices on half of a planform

NUT(1) — Number of points on one side of the x-z symmetry plane at which influence coefficients have been computed in Program A when ISOLVE = 1 (NUT(1) = NVORT)

NUT(2) — Number of influence coefficient points when ISOLVE = 2. The contribution of NUT(2) from each wing part, IW, is NSPAN(IW) x (NCHORD(IW) - 1).

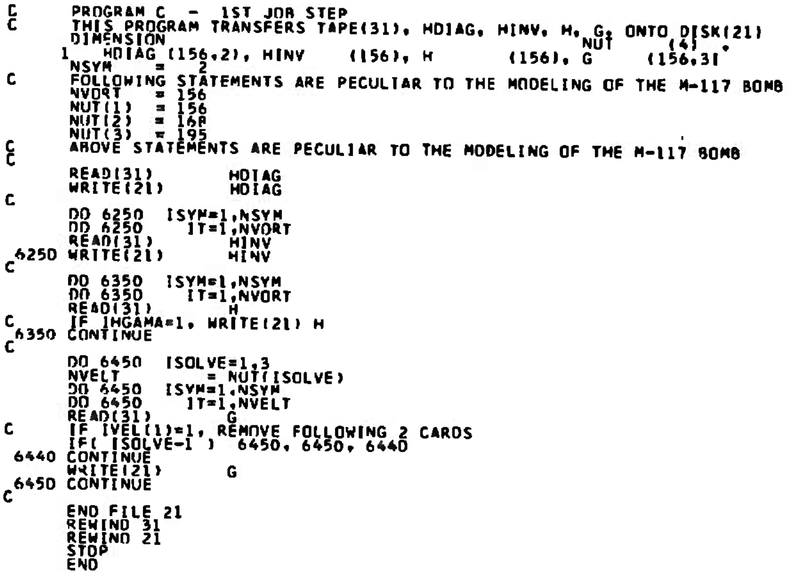
NUT(3) — Number of influence coefficient points when ISOLVE = 3. The contribution to NUT(3) from each wing part, IW, is (NSPAN(IW) - 1) x NCHORD(IW).

FORTRAN statements used to compute NUT(1), NUT(2), and NUT(3) are to be found in Programs A and B, and in the second job step of Program C.

The matrices of geometric factors which were written in Program B as a data set using reference number 21 are read using reference number 31 in this job step.

Dimensions of Arrays

Wherever the number 156 appears as a dimension in the program listing, it indicates that the dimensions in those locations are equal to NVORT.



```
C       PROGRAM C  -  1ST JOB STEP
C       THIS PROGRAM TRANSFERS TAPE(31), HDIAG, HINV, H, G, ONTO DISK(21)
        DIMENSION                                              NUT        (4) ,
      1    HDIAG (156,2), HINV     (156), H         (156), G       (156,3)
        NSYM     =   2
C       FOLLOWING STATEMENTS ARE PECULIAR TO THE MODELING OF THE M-117 BOMB
        NVORT    = 156
        NUT(1)   = 156
        NUT(2)   = 168
        NUT(3)   = 195
C       ABOVE STATEMENTS ARE PECULIAR TO THE MODELING OF THE M-117 BOMB
C
        READ(31)          HDIAG
        WRITE(21)         HDIAG
C
        DO 6250   ISYM=1,NSYM
        DO 6250     IT=1,NVORT
        READ(31)          HINV
 6250   WRITE(21)         HINV
C
        DO 6350   ISYM=1,NSYM
        DO 6350     IT=1,NVORT
        READ(31)          H
C       IF IHGAMA=1, WRITE(21) H
 6350   CONTINUE
C
        DO 6450   ISOLVE=1,3
        NVELT           = NUT(ISOLVE)
        DO 6450   ISYM=1,NSYM
        DO 6450     IT=1,NVELT
        READ(31)          G
C       IF IVEL(1)=1, REMOVE FOLLOWING 2 CARDS
        IF( ISOLVE-1 )  6450, 6450, 6440
 6440   CONTINUE
        WRITE(21)         G
 6450   CONTINUE
C
        END FILE 21
        REWIND 31
        REWIND 21
        STOP
        END
```

Output Data

A subset of the data set containing HDIAG, H^{-1}, H, and $\vec{G}$, which is read from tape in this job step, is written on magnetic disk, data set reference number 21. See Section 3. 3. 1 for details.

2.4 USER'S GUIDE TO PROGRAM C—SECOND JOB STEP

Input Data

The input data are similar to those of Program A, except for the additions noted below.

The following data are read from punched cards:

IHGAMA = 0 H is not contained on data set reference number 21
= 1 H is contained on data set reference number 21

ICF = 1 Laminar skin friction assumed
= 2 Turbulent skin friction assumed

NSCORD(IW) Number of chordwise network points on the geometric surface of wing part IW; force coefficients are summed only over vortex segments corresponding to these points (e.g., not over the segments in the wake of the sample run described herein)

XM, ZM Body axis coordinates of pitch axis, in.

YL, ZL Body axis coordinates of roll axis, in.

XN, YN Body axis coordinates of yaw axis, in.

XCG, YCG, ZCG Body axis coordinates of center of gravity, in.

IBC = 0 Store is assumed immersed in a uniform flow field; i. e., no parent aircraft is present
= 1 Store is assumed immersed in a nonuniform flow field

IPRVFS = 0 Do not print the NUFF velocity components in SUBROUTINE FRESTR
= 1 Print the NUFF velocity components

IPRVEL = 0 Do not print velocity distributions in SUBROUTINE VELOCY
= 1 Print velocity distributions

IPRGAM = 0 Do not print the strengths of the vortices, Γ_j
= 1 Print Γ_j

IPRCP = 0 Do not print the pressure coefficient, C_p, distribution
= 1 Print C_p

IPRCF = 0 Do not print the force coefficient distribution
= 1 Print the force coefficient

IX1, IX2 Counters for the initial and final grid points at which NUFF data are input in the x-direction, usually IX1 = 1

IY1, IY2 Counters for initial and final NUFF grid points in the y-direction, usually IY1 = 1

IZ1, IZ2 Counters for initial and final NUFF grid points in the z-direction, usually IZ1 = 1

DOWNV (IX, IY, IZ) Downwash angles (deg) of the NUFF (positive DOWNV is upwash)

SIDEV (IX, IY, IZ) Sidewash angles (deg) of the NUFF (positive SIDEV is inwash on negative y-side of the x-z plane of symmetry)

VMAG (IX, IY, IZ) Ratio of the magnitude of the local velocity vector to the magnitude of the velocity at infinity

The following data are defined by FORTRAN arithmetic expressions:

XP(IX), YP(IY) ZP(IZ) — Grid points at which NUFF data is input These points must be ordered in increasing values of XP, YP, and ZP

S — Reference area used in definition of the force coefficients of a model store, in.^{2}

CBAR — Reference chord length used in definition of pitching- and yawing-moment coefficients of a model store, in.

BSPAN — Reference lateral dimension used in definition of rolling-moment coefficient of a model store, in.

XNET, YNET, ZNET — Coordinates of the vortex network points; these must be the same as the corresponding values in Program A

Dimensions of Arrays

The dimensions in this job step are the same as those in Program A, with the addition of the following. The counters for the number of grid points in the X, Y, and Z directions at which the nonuniform flow field is specified (IX2, IY2, and IZ2) are the dimensions of the coordinates of these points. These counters also constitute the dimensions of the downwash, sidewash, and velocity magnitude variables at these grid points. Thus, the DIMENSION statement is based upon the values XP(IX2), YP(IY2), and ZP(IZ2); and DOWNV(IX2, IY2, IZ2), SIDEV (IX2, IY2, IZ2), and VMAG(IX2, IY2, IZ2).

PROGRAM C
SECOND JOB STEP
INPUT DATA FROM PUNCHED CARDS
SAMPLE RUN

```
FORCE
COEFF    IBC=0   M=0.

        AMACH
         0.0

       IHGAMA                  ICF
            0                    1

          NWP
            3

      NCHORD( 1)          NSPAN(1)            NCHORD(2)            NSPAN(2)            NCHORD(3)            NSPAN(3)
           24                    5                    9                   5                    9                   5

      NSCORD( IW)
           20                    9                    9

        MTIP( 1)          MROOT(1)             MTIP(2)            MROOT(2)              MTIP(3)            MROOT(3)
            0                    0                    1                   1                    1                   1

          XCG                  YCG                  ZCG
      -1.6440             0.0                 0.0

           XM                   ZM                   YL                  ZL                   XN                  YN
      -1.6440             0.0                 0.0                 0.0                 -1.6440              0.0

            S                BSPAN                 CBAR
       0.5027             0.8000               4.3950

         NSYM                  IBC
            2                    0

       IVEL( 1)             IVEL(2)              IVEL(3)             IVEL(4)
            1                    1                    1                   0

         IPRG     IPRVFS     IPRVEL     IPRGAM      IPRCP      IPRCF
            0          0          0          0          1          1

          IX1                  IX2                  IY1                 IY2                  IZ1                 IZ2
            1                   10                    1                   8                    1                  10
```

```
C     PROGRAM C  -  2ND JOB STEP
C     THIS JOB STEP PERFORMS
C        1) READ(5,     )  INPUT DATA AND APPLIES GOETHERT TRANSFORMATION
C        2) COMPUTE  COMMON/CNET/  AND  COMMON/CCOORD/
C        3) WRITE(51)       INPUT DATA, COMMON/CNET/, COMMON/CCOORD/
C        4) PASSES HDIAG, HINV, G ON DISK UNIT(21) TO THE NEXT JOB STEP
      COMMON/CNET  /     NWP           , NIWP          ,
     1   NCHORD(3)     , NSPAN (3)     , NSCORD(3)     , NVOR  (3)     ,
     2   NUC    (3,4)  , NUS   (3,4)   , NVEL  (3,4)   , NUT       (4)
      COMMON/CCOORD/     XNET (24,5,3), YNET (24,5,3), ZNET (24,5,3),
     1                   AKSI (23,4,3), ETA  (23,4,3), ZETA (23,4,3),
     2                   XFLOWF( 1, 1), YFLOWF( 1, 1), ZFLOWF( 1, 1),
     3   BOC    (156,3), AX          (3), AY          (3), AZ          (3)
      DIMENSION
     1   MTIP(3)       , MROOT(3)      , IVEL(4)       ,
     7   XP(10)        , YP(8)         , ZP(10)        ,
     8   DOWNV(10,8,10),SIDEV(10,8,10),VMAG(10,8,10)
      DIMENSION          CXD         (3)
C
C
C     XP(IX), YP(IY), ZP(IZ)  NONUNIFORM FLOW FIELD GRID POINTS
C                             IN PARENT A/C REF. SYSTEM
C     XORIG , YORIG , ZORIG   ROTATIONAL AXES COORDINATES
C                             IN PARENT A/C REF. SYSTEM
C     XM, ZM, YL, ZL, XN, YN  ROTATIONAL AXES COORDINATES
C                             IN STORE AXES REF. SYSTEM
C     XVP    , YVP    , ZVP   POINTS AT WHICH VELOCITY IS COMPUTED
C                             IN PARENT A/C REF. SYSTEM
C     XCG    , YCG    , ZCG   STORE CENTER OF GRAVITY
C                             IN STORE AXES REF. SYSTEM
 7100 FORMAT( 16I5    )
 7200 FORMAT( 8E10.0 )
 7250 FORMAT( 10E8.0 )
 7300 FORMAT( 80H
     1                                     )
      READ(5,7300)
      READ(5,7200)        AMACH
      READ(5,7100)        IHGAMA,    ICF
      READ(5,7100)        NWP
      READ(5,7100)       (NCHORD(IW),      NSPAN(IW), IW=1,NWP)
      READ(5,7100)       (NSCORD(IW),                IW=1,NWP)
      READ(5,7100)       (MTIP(IW),        MROOT(IW), IW=1,NWP)
      READ(5,7200)        XCG    ,    YCG    ,   ZCG
      READ(5,7200)        XM,   ZM,   YL,   ZL,   XN,   YN
      READ(5,7100)        NSYM   ,    IBC
      READ(5,7100)        IVEL(1),   IVEL(2),   IVEL(3),   IVEL(4)
      READ(5,7100)        IPRG   ,
     1                    IPRVFS,    IPRVEL,    IPRGAM,    IPRCP ,    IPRCF
C     NON-UNIFORM FREE STREAM BOUNDARY CONDITIONS
      READ(5,7100)        IX1, IX2, IY1, IY2, IZ1, IZ2
C     FORMULAS FOR XP(IX), YP(IY), ZP(IZ) ARE PECULIAR TO M-117 BOMB
      DO 1110       IX=IX1,IX2
 1110 XP(IX)          = -18. + IX - IX1
      DO 1120       IY=IY1,IY2
 1120 YP(IY)          = - 7. + IY - IY1
      DO 1130       IZ=IZ1,IZ2
 1130 ZP(IZ)          = FLOAT( IZ )
      DO 1140       IY=IY1,IY2
      DO 1140       IZ=IZ1,IZ2
 1140 READ(5,7250)        (DOWNV(IX,IY,IZ), IX=IX1,IX2)
      DO 1150       IY=IY1,IY2
      DO 1150       IZ=IZ1,IZ2
 1150 READ(5,7250)        (SIDEV(IX,IY,IZ), IX=IX1,IX2)
      DO 1160       IY=IY1,IY2
      DO 1160       IZ=IZ1,IZ2
 1160 READ(5,7250)        (VMAG(IX,IY,IZ), IX=IX1,IX2)
C
C
C
C     FOLLOWING STATEMENTS ARE PECULIAR TO THE MODELING OF THE M-117 BOMB
      BSPAN          =  .8
      S              = 3.1416 * (BSPAN/2.)**2
      CBAR           = 4.395
C
C
      XNET( 5,1,1) = 4.395
      XNET( 9,1,1) = 3.435
      XNET(11,1,1) = 2.935
      XNET(15,1,1) = 2.035
      XNET(19,1,1) = 1.058
      XNET(24,1,1) = 0.
C
      DO 1254       I=1,8
 1254 XNET(I,1,1)    = XNET(5,1,1) +  ( I-5) * (XNET(11,1,1)-XNET(5,1,1))
     1                               / (11-5)
      XNET(10,1,1) = ( XNET(9,1,1) + XNET(11,1,1) )  /  2.
C
      DO 1256       I=12,14
 1256 XNET(I,1,1)    = XNET(11,1,1)+  ( I-11)* (XNET(15,1,1)-XNET(11,1,1))
     1                               / (15-11)
C
      DO 1258       I=16,18
 1258 XNET(I,1,1)    = XNET(15,1,1)+  ( I-15)* (XNET(19,1,1)-XNET(15,1,1))
     1                               / (19-15)
C
      DO 1260       I=20,23
 1260 XNET(I,1,1)    = XNET(19,1,1)+  ( I-19)* (XNET(24,1,1)-XNET(19,1,1))
     1                               / (24-19)
      NNC            = NCHORD(1)
      NNS            = NSPAN(1)
      DO 1262       I=1,NNC
      DO 1262       J=2,NNS
 1262 XNET(I,J,1)    = XNET(I,1,1)
C
      XO             = 1.055
      ZO             = - SQRT( 1.6**2 - XO**2 )
      ZNET( 1,1,1) = -.2815
      ZNET(11,1,1) = -.2815
      ZNET(15,1,1) = -.4
      ZNET(19,1,1) = -.4
      ZNET(24,1,1) = 0.
C
      DO 1294       I=1,NNC
      IF( I-11 )  1282, 1282, 1272
 1272 IF( I-15 )  1284, 1284, 1274
 1274 IF( I-19 )  1286, 1286, 1276
 1276 IF( I-24 )  1288, 1290, 1290
C
C     I=1,11
 1282 ZCL            = - ZNET(1,1,1)
```

```
      GO TO 1292
C
C     I=12,15
 1284 ZCL            = ZNEI(11,1,1)+, ( I-11)* IZNET(15,1,1I-ZNET(11,1,1)I
     1                                   / (15-11)
      ZCL            = - ZCL
      GO TO 1292
C
C     I=16,19
 1286 ZCL            = - ZNET(15,1,1)
      GO TO 1292
C
C     I=20,23
 1288 ZCL            =    ZO
     1                 + SORT( 1.6**2 - (XNET(I,I,1)-XOI**2 )
      GO TO 1292
C
C     I=24
 1290 ZCL            = 0.
C
 1292 YNET(1,1,1)    = 0.
      YNET(I,2,1)    = ZCL * SIN( 3.14I6 / 4. I
      YNET(I,3,1)    = ZCL
      YNET(I,4,1)    = ZCL * SIN( 3.1416 / 4. I
      YNET(I,5,1)    = 0.
      ZNET(1.1,1)    = - ZCL
      ZNEI(I,2,1)    = - ZCL * SIN( 3.1416 / 4. I
      ZNET)I.,3,1)   =   0.
      ZNET)I,4,1)    =   ZCL * SIN( 3.1416 / 4. I
      ZNET(I,5,1)    =   ZCL
 1294 CONTINUE
      IF( NWP-I I  1380, .1380, 1300
 I300 NNC   = NCHORD(2)
      NNCM1 = NCHORO(2)-1
      NNS   = 'NSPAN(2)
      NNSM1 = NSPAN(2) -1
      XNET(1   ,1   ,2) =  3.9450
      XNET(1   ,NNS,2) =  3.4350
      XNET(NNC,1   ,2) =  4.3950
      XNET(NNC,NNS,2) =  4.3950
      DO 1312            I=1,NNC
      XNET(I   ,1   ,2) = XNET(1,1   ,2) + ( XNET(NNC,1.,2I-XNET(.1,1,2I I
     1                     * FLDATII-1) / (NNC-1I
      XNET(I   ,NNS,2) = XNET(1,NNS,2)
     1                    +( XNET(NNC,NNS,2)-XNET(1,NNS,2) I
     2                     * FLOAT(I-1) '/ (NNC-1I
         DO 1310         J=1,NNS
 1310    XNET(I,J,2)   = XNET(I,1,2)   + I XNET(I,NNS,2)-XNET(I,1,2I I
     1                     * FLOAT(J-1) ./ (NNS-1)
 1312 CONTINUE
C
      YNET(1,1   ,2)=  0.4020
      YNET(1,NNS,2)=  0.1991
      ZNET(1,1   ,2)= -YNET(1.1   ,2)
      ZNET(1,NNS,2)= -YNET(1,NNS,2)
      DO 1350       J=1,NNS
      YNET(I,J,2)   =     YNET(1,1,2)  '+   ( YNET(1,NNS,2I - YNETI1,1,2I I
     1                  * FLOAT(J-1) / (NNS-1)
 1350 ZNET(1,J,2)   =     ZNET(1,1,2)   +   ( ZNET(1,NNS,2) - ZNET(1,1,2I I
     1                  * FLDAT(J-1) / '(NNS-1I
C
      00 1360       I=2,NNC
      DO 1360       J=1,NNS
      YNET(I,J,2)   = YNET(1,J,2)
 1360 ZNET(I,J,2)   = ZNET(1,J,2)
C
      DD 1370       I=1,NNC
      OO 1370       J=1,NNS
      XNET(I,J,3)   =  XNET(I,J,2)
      YNET(I,J,3I   =  YNET(I,J,2)
 1370 ZNET(I,J,3)   = -ZNET(I,J,2)
 13B0 CONTINUE
C
      DO 1420      IW=1,NWP
      NNC              = NCHORO(IW)
      NNS              = NSPAN(IW)
      DD 1420         I=1,NNC
      DO 1420         J=1,NNS
'     XNET(I,8,IW) = - XNET(I,J,IW)
 1420 YNET(I,J,IW) = - YNET(I,J,IW)
C     ABOVE STATEMENTS ARE PECULIAR TO THE MODELING OF THE M-117 BOMB
C
C
C
      BETA             = SORT( 1. - AMACH**2 )
      S                = BETA**2 * S
      BSPAN            = BETA * BSPAN
      YCG              = BETA * YCG
      ZCG              = BETA * ZCG
      YL               = BETA * YL
      YN               = BETA * YN
      ZL          .    = BETA * ZL
      ZM               = BETA * ZM
      DO 1500      IW=1,NWP
      NNC              = NCHORD(IW)
      NNS              = NSPAN()W)
      DO 1500       I=I,NNC
      DO 1500       J=1,NNS
      YNET(I,J,IW) = BETA * YNET(I,J,IW)
 1500 ZNET(I,J,IW) = BETA * ZNET(I,J,IW)
C
C
      DO 1520      IY=IY1,IY2
 1520 YP(IYI           = BETA * YP(IY)
      DO 1530      IZ=IZ1,IZ2
 1530 ZP(IZ)           = BETA * ZP(IZ)
C
C
      DO 1550      IX=IX1,IX2
      DO 1550      IY=IY1,IY2
    . DO 1550       IZ=IZ1,IZ2
      DOWNV(IX,IY,IZ) = BETA * DOWNV(IX,IY,IZI
 1550 SIDFV(IX,IY,IZ) = BETA * SIDEV(IX,IY,IZI
C
      WRITE(6,7300)
      WRITE(6,8004)                          'AMACH
 8004 FORMAT(/         '                      AMACH                    '/6F20.2I
      WRITE(6,8006)                          IHGAMA,                ICF
 8006 FORMAT(/         '                      IHGAMA                 ICF'/6I20   I
      WRITE(6,8010)                           NWP
```

```
8010 FORMAT(/            '                    NWP                          '/6I2D  )
     WR(TE(6,8020)                        (NCHORD(IWI,              NSPAN(IWI,
     I                                      1W=I,NWPI
8020 FORMAT(/            '                  NCHORD(I)                NSPAN(II
     1         NCHORD(2)              NSPAN(7)               NCHORD(3I
     2NSPAN(3)   '                                    '                   /6I20  )
     WRITF(6,8030)                        (NSCORD(IW), IW=I,NWPI
8D30 FORMAT(/            '                  NSCORD(IWI                 '/6I2D  I
     WRITF(6,8O4D)                          (MTIP(IWI,               MROOT(IW),
     I                                       IW=1,NWP)
8D40 FORMAT(/            '                  MT)P(II                  MROOT(I)
     1            MTIP(2)             MROOT(2)               MTIP(3)
     2MROOT(3)  '                                                     /6I20  I
     WRITE(6,806D)                          XCG,                   YCG,
     1            ZCG
8D60 FORMAT(/            '                  XCG                    YCG
     1            ZCG                                                '/6F20.4)
     WRITE(6,8050)                           XM,   ZM,   YL,   ZL,  XN,   YN
8D50 FORMAT(/            '                   XM                     7M
     1              YL                ZL                  XN
     2    YN        '                                                 /6F70.4I
     WRITF(6,8070)                            S,                   BSPAN,
     I  CBAR
8D70 FORMAT(/            '                    S                    BSPAN
     I           CBAR                                              '/6F20.4I
     WRITE(6,8080)                          NSYM,                  IBC
8080 FORMAT(/            '                  NSYM                   IBC'/6I2D  I
     WRITE(6,8090)                        (IVEL(ISDLVEI, ISOLVE=1,4I
8D9D FORMAT(/            '                  IVEL(1I                IVFL(2)
     1           IVEL(3)              IVEL(4)                      '/6I20  I
     WRITF(6,8100)       '                  IPRG,
     1                 IPRVFS,     IPRVEL,     IPRGAM,     IPRCP,      IPRCF
8100 FORMAT(/            '                  IPRG',
     1          '      IPRVFS      IPRVEL      IPRGAM      IPRCP       IPRCF'/
     2                                                          IOX, 12I10I
     WRITE(6,8110)                            IXI, IX2, IYI, IY2, IZI, IZ2
8110 FORMAT(/            '                    IXI                 IX2
     1           IYI                  IY2                  IZ1
     2    IZ2'                                                      /6I2D  )
C
C
     DO 2130   IW=1,NWP
     NUC(IW,1)   = NCHORD(IW)-1
     NUS(IW,1)   = NSPAN(IW) -1
     NUC(IW,2)   = NCHORO(IW)
     NUS(IW,2)   = NSPAN(1W) -1
     NUC(IW,3)   = NCHORD(IW)-1
2130 NUS(IW,3)   = NSPAN(IW)
C
     NVOR( 1)           =                     (NCHORD( 1I-1) * (NSPAN( 1)-I)
     OO 2140    ISOLVE=I,3
2140 NVEL( 1,ISOLVE) =    NUC( 1,ISOLVE) * NUS( 1,ISOLVE)
     IF( NWP-1 )  2158, 2158, 2148
2148 DO 2150          IW=2,NWP
     NVOR(IW)           =    NVOR(IW-I)   +  (NCHORD(IWI-1) * (NSPAN(IWI-1)
     DO 2150    ISOLVE=1,3
2150 NVEL(IW,ISOLVE) =    NVEL(IW-1,(SOLVE)
    1                +    NUC(IW,ISOLVE) * NUS(IW,ISOLVE)
2158 OO 2160    ISOLVE=1,3
216D NUT((SOLVE)        =    NVEL( NWP,ISOLVE)
     NVORT              =    NVOR(NWP)
C
     OO 2670   IW=I,NWP
     NVORC         = NCHORO(IW)-1
     NVORS         = NSPAN(IW) -1
     DO 2650   IP=1,NVORC
     I             = )P
     DO 2650   IO=1,NVORS
     J             = IO
     IF( IW-1 )  2610, 2610, 2620
2610 IT            =                 (IP-1)*NVORS + IO
     GO TO 2630
2620 IT            = NVOR(IW-1) + (IP-1)*NVORS + IO
2630 CONT(NUE
     AKSI(IP,IO,IW) = (   XNET(I   ,J,IW) + XNET(I   ,J+1,IW)
    1                   + XNET(1+1,J,IW) + XNET(]+1,J+1,IW)   I / 4.
     ETA (IP,(O,1W) = (   YNET(I   ,J,IW) + YNET(I   ,J+1,IW)
    I                   + YNET(I+I,J,IW) + YNET(I+1,J+1,IW)   I / 4.
     ZETA(1P,IO,IW) = (   ZNET(I   ,J,IW) + ZNET(I   ,J+1,1W)
    1                   + ZNET(I+1,J,IW) + ZNET(I+1,J+1,IW)   I / 4.
     CX            = XNET(I,J+I,)W) - XNET()+I,J,IWI
     CY            = YNET(I,J+I,IW) - YNET(I+1,J,IWI
     CZ            = ZNET(I,J+1,IW) - ZNET(1+1,J,IWI
     OX            = XNET(I,J   ,IW) - XNET(I+1,J+1,IWI
     DY            = YNET(1,J   ,IW) - YNET(I+1,J+1,IW)
     OZ            = ZNET(1,J   ,1W) - ZNET(I+1,J+1,IW)
     CXD(1)        =    CY * DZ - OY * CZ
     CXO(2)        = - CX * DZ + OX * CZ
     CXD(3)        =    CX * DY - OX * CY
     ABSCXD        = SORT( CXD(1)**2 + CXO(2)**2 + CXD(3I**2 I
     DO 2650    N=1,3
2650 BDC(IT,N)   = CXD(N) / ABSCXD
C
2670 CONTINUE
C
     WRITE(6,8506)
8506 FORMAT('1', 2X  '        CONTROL VOLUME GR(D POINTS.  NONUNIFORM FLO
    1W FIELD.          ' )
     WRITE(6-8510)               (XP((X), 1X=IX1,IX2I
8510 FORMAT(      3X, '           XP(IX)=', 10F10.5 )
     WRITE(6,8520)               (YP(IY), IY=IY1,IY2)
8520 FORMAT(      3X, '           YP(1Y)=', 10F10.5 I
     WRITE(6,8530)               (ZP(IZ), IZ=IZ1,IZ2)
853D FORMAT(      3X, '           ZP(IZ)=', 10F10.5 )
     OO 8616       IX=IX1,IX2
     WRITE(6,8610)       IX,     XP(IX)
8610 FORMAT(/     3X  '           DOWNWASH (DEG) AT PARENT AIRCRAFT     X-STA
    1      IX=',I2,   '           XP(IX)=', F10.5I
        DO 8612 IZ=IZ1,IZ2
8612    WRITE(6,8614)   IZ,   ZP(IZ),  (DOWNV(IX,IY,IZI, IY=1Y1,IY2I
8614    FORMAT(        ' IZ=',12, 2X, ' ZP(IZI=',F10.5,      I5X,
    I                                  '(DOWNV, 1Y=1Y1,IY2I =',8F7.2      I
8616 CONTINUE
C
     DO 8626       IX=IX1,IX2
     WR1TE(6,8620)       IX,     XP(IX)
8620 FORMAT(/     3X, '           SIDEWASH (OEG) AT PARENT AIRCRAFT     X-STA
    1      IX=',I2,   '           XP(IX)=', F10.5 I
        OO 8622 IZ=IZ1,IZ2
8622    WRITE(6,8624)   IZ,   ZP(IZ),  (SIDEV(IX,IY,IZI, IY=IY1,IY2I
 8624     FORMAT(         ' IZ=',I2, 2X, ' ZP(IZ)=',F10.5,       15X,
     1                                   '(SIDEV, IY=IY1,IY2) =',8F7.2      I
 8626 CONTINUE
C
      DO 8636     IX=IX1,IX2
      WRITE(6,8630)      IX,  XP(IX)
 8630 FORMAT(/     3X, '      VELOCITY MAGNITUDE AT PARENT AIRCRAFT X-STA
     1     IX=',I2,    '      XP(IX)=', F10.5 )
          DO 8632 IZ=IZ1,IZ2
 8632     WRITE(6,8634)  IZ,  ZP(IZ),  (VMAG (IX,IY,IZ), IY=IY1,IY2)
 8634     FORMAT(         ' IZ=',I2, 2X, ' ZP(IZ)=',F10.5,       15X,
     1                                   '(VMAG,  IY=IY1,IY2) =',8F7.4      I
 8636 CONTINUE
C
      DO 8290  IW=1,NWP
      NNC          = NCHORD(IW)
      NNS          = NSPAN(IW)
      GO TO( 8210, 8220, 8230), IW
 8210 WRITE(6,8212)  IW
 8212 FORMAT(//  ' NETWORK POINTS ON BODY            IW=',I1,
     1   '        J=1 IS TOP (-Z) CL        J=NJ IS BOTTOM (+Z) CL 'I
      GO TO 8250
 8220 WRITE(6,8222)  IW
 8222 FORMAT(//  ' NETWORK POINTS ON UPPER FIN   IW=',I1,
     1   '        J=1 IS TIP                J=NJ IS ROOT              ')
      GO TO 8250
 8230 WRITE(6,8232)  IW
 8232 FORMAT(//  ' NETWORK POINTS ON LOWER FIN   IW=',I1,
     1   '        J=1 IS TIP                J=NJ IS ROOT              ')
 8250 WRITE(6,8260)
 8260 FORMAT(/        ' XNET(I,J,IW)      ')
      DO 8262    I=1,NNC
 8262 WRITE(6,8264)  I, (XNET(I,J,IW), J=1,NNSI
 8264 FORMAT(         ' I=',I2,  9F10.4     )
      WRITE(6,8270)
 8270 FORMAT(/        ' YNET(I,J,IW)      ')
      DO 8272    I=1,NNC
 8272 WRITE(6.8274)  I, (YNET(I,J,IW), J=1,NNSI
 8274 FORMAT(         ' I=',I2,  9F10.4     )
      WRITE(6,8280)
 8280 FORMAT(/        ' ZNET(I,J,IW)      ')
      DO 8282    I=1,NNC
 8282 WRITE(6,8284)  I, (ZNET(I,J,IW), J=1,NNS)
 8284 FORMAT(         ' I=',I2,  9F10.4     )
 8290 CONTINUE
C
C
      DO 8388  IW=1,NWP
      NVORC        = NCHORD(IW)-1
      NVORS        = NSPAN(IW) -1
      WRITE(6,8310)  IW
 8310 FORMAT(//       ' BOUNDARY POINTS              IW=',I1            )
      WRITE(6,8360)
 8360 FORMAT(/        ' AKSI(IP,IO,IW) '  )
      DO 8362  IP=1,NVORC
 8362 WRITE(6,8364)  IP, (AKSI(IP,IO,IW), IO=1,NVORS)
 8364 FORMAT(         ' IP=',I2, 8F10.5     )
      WRITE(6,8370)
 8370 FORMAT(/        '  ETA(IP,IO,IW) '  )
      DO 8372  IP=1,NVORC
 8372 WRITE(6,8374)  IP, ( ETA(IP,IO,IW), IO=1,NVORS(
 9374 FORMAT(         ' IP=',I2, 8F10.5     )
      WRITE(6,8380)
 8380 FORMAT(/        ' ZETA(IP,IO,IW) '  )
      DO 8382  IP=1,NVORC
 8382 WRITF(6,8384)  IP, (ZETA(IP,IO,IW), IO=1,NVORS)
 8384 FORMAT(         ' IP=',I2, 8F10.5     )
 8388 CONTINUE
      WRITE(6,8410)
 8410 FORMAT(//       ' BDC(IT,N)     DIRECTION COSINES AT BOUNDARY POINTS
     1      N=1 IS X-COMPONENT        N=2 IS Y      N=3 IS Z   'I
      DO 8420      IW=1,NWP
      NVORC           = NCHORD(IW)-1
      NVORS           = NSPAN(IW) -1
      DO 8420        N=1,3
      DO 8420       IP=1,NVORC
      IF( IW-1 )   8412, 8412, 8414
 8412 IT1             =                 (IP-1)*NVORS + 1
      IT2             =                  IP   *NVORS
      GO TO 8416
 8414 IT1             = NVOR(IW-1) + (IP-1)*NVORS + 1
      IT2             = NVOR(IW-1) +  IP   *NVORS
 8416 WRITE(6,8418)  IW, N, IP,    (BDC(IT,NI, IT=IT1,IT2)
 8418 FORMAT(         ' IW=',I1, '    N=',I1, '    IP=',I2,  8F10.5 )
 8420 CONTINUE
C
C
      WRITE(51)
     1    AMACH          ,
     2    IHGAMA         , ICF           ,
     3    NWP            ,
     4    NCHORD         , NSPAN         ,
     5    NSCORD         ,
     6    MTIP           , MROOT         ,
     7    XCG            , YCG           , ZCG          ,
     8    XM       , ZM  ,     YL        , ZL     , XN        , YN       ,
     9    S              , BSPAN         , CBAR         ,
     A    NSYM           , IRC           ,
     A    IVEL           ,
     B    IPRG   ,   IPRVFS,  IPRVEL,    IPRGAM,   IPRCP ,   IPRCF ,
     D    IX1      , IX2     , IY1       , IY2    , IZ1       , IZ2      ,
     E    XP             , YP            , ZP           ,
     F    DOWNV          , SIDEV         , VMAG
      WRITE(51)                                          NVOR        ,
     1    NUC            , NUS           , NVEL       , NUT         ,
     2    XNET           , YNET          , ZNET       ,
     3    AKSI           , ETA           , ZETA       ,
     4    BDC
      END FILE 51
      REWIND 51
C
      STOP
      END
```

NETWORK POINTS ON BODY

```
                         IW=1          J=1 IS TOP I-Z1 CL
                                       J=NJ IS BOTTOM I+Z1 CL
XNET(I,J,IW)
I= 1    -5.3683   -5.3683   -5.3683   -5.3683   -5.3683
I= 2    -5.1250   -5.1250   -5.1250   -5.1250   -5.1250
I= 3    -4.8817   -4.8817   -4.8817   -4.8817   -4.8817
I= 4    -4.6383   -4.6383   -4.6353   -4.6353   -4.6383
I= 5    -4.3950   -4.3950   -4.3950   -4.3950   -4.3950
I= 6    -4.1517   -4.1517   -4.1517   -4.1517   -4.1517
I= 7    -3.9053   -3.9083   -3.9093   -3.9083   -3.9053
I= 8    -3.6650   -3.6650   -3.6650   -3.6650   -3.6650
I= 9    -3.4350   -3.4350   -3.4350   -3.4350   -3.4350
I=10    -3.1550   -3.1850   -3.1550   -3.1550   -3.1850
I=11    -2.9350   -2.9350   -2.9350   -2.9350   -2.9350
I=12    -2.7100   -2.7100   -2.7100   -2.7100   -2.7100
I=13    -2.4850   -2.4850   -2.4850   -2.4850   -2.4850
I=14    -2.2600   -2.2600   -2.2600   -2.2600   -2.2600
I=15    -2.0350   -2.0350   -2.0350   -2.0350   -2.0350
I=16    -1.7907   -1.7907   -1.7907   -1.7907   -1.7907
I=17    -1.5465   -1.5465   -1.5465   -1.5465   -1.5465
I=18    -1.3022   -1.3022   -1.3022   -1.3022   -1.3022
I=19    -1.0580   -1.0580   -1.0550   -1.0580   -1.0580
I=20    -0.8464   -0.8464   -0.8464   -0.8464   -0.8464
I=21    -0.6348   -0.6348   -0.6345   -0.6348   -0.6348
I=22    -0.4232   -0.4232   -0.4232   -0.4232   -0.4232
I=23    -0.2116   -0.2116   -0.2116   -0.2116   -0.2116
I=24     0.0       0.0       0.0       0.0       0.0
```

```
YNET(I,J,IW)
I= 1     0.0      -0.1991   -0.2815   -0.1991    0.0
I= 2     0.0      -0.1991   -0.2815   -0.1991    0.0
I= 3     0.0      -0.1991   -0.2815   -0.1931    0.0
I= 4     0.0      -0.1991   -0.2915   -0.1991    0.0
I= 5     0.0      -0.1991   -0.2815   -0.1991    0.0
I= 6     0.0      -0.1991   -0.2515   -0.1991    0.0
I= 7     0.0      -0.1991   -0.2815   -0.1991    0.0
I= 8     0.0      -0.1991   -0.2815   -0.1991    0.0
I= 9     0.0      -0.1991   -0.2815   -0.1991    0.0
I=10     0.0      -0.1991   -0.2915   -0.1991    0.0
I=11     0.0      -0.1991   -0.2815   -0.1991    0.0
I=12     0.0      -0.2200   -0.3111   -0.2200    0.0
I=13     0.0      -0.2409   -0.3407   -0.2409    0.0
I=14     0.0      -0.2619   -0.3704   -0.2619    0.0
I=15     0.0      -0.2828   -0.4000   -0.2525    0.0
I=16     0.0      -0.2828   -0.4000   -0.2828    0.0
I=17     0.0      -0.2828   -0.4000   -0.2828    0.0
I=18     0.0      -0.2528   -0.4000   -0.2528    0.0
I=19     0.0      -0.2528   -0.4000   -0.2828    0.0
I=20     0.0      -0.2711   -0.3834   -0.2711    0.0
I=21     0.0      -0.2411   -0.3409   -0.2411    0.0
I=22     0.0      -0.1888   -0.2671   -0.1888    0.0
I=23     0.0      -0.1108   -0.1565   -0.1108    0.0
I=24     0.0       0.0       0.0       0.0       0.0
```

```
ZNET(I,J,IW)
I= 1    -0.2815   -0.1991    0.0       0.1991    0.2815
I= 2    -0.2815   -0.1991    0.0       0.1991    0.2815
I= 3    -0.2815   -0.1991    0.0       0.1991    0.2515
I= 4    -0.2815   -0.1991    0.0       0.1991    0.2815
I= 5    -0.2815   -0.1991    0.0       0.1991    0.2815
I= 6    -0.2815   -0.1991    0.0       0.1991    0.2815
I= 7    -0.2815   -0.1991    0.0       0.1991    0.2815
I= 8    -0.2815   -0.1991    0.0       0.1991    0.2815
I= 9    -0.2815   -0.1991    0.0       0.1991    0.2815
I=10    -0.2815   -0.1991    0.0       0.1991    0.2815
I=11    -0.2815   -0.1991    0.0       0.1991    0.2815
I=12    -0.3111   -0.2200    0.0       0.2200    0.3111
I=13    -0.3407   -0.2409    0.0       0.2409    0.3407
I=14    -0.3704   -0.2619    0.0       0.2619    0.3704
I=15    -0.4000   -0.2828    0.0       0.2828    0.4000
I=16    -0.4000   -0.2825    0.0       0.2828    0.4000
I=17    -0.4000   -0.2828    0.0       0.2828    0.4000
I=18    -0.4000   -0.2825    0.0       0.2825    0.4000
I=19    -0.4000   -0.2825    0.0       0.2928    0.4000
I=20    -0.3834   -0.2711    0.0       0.2711    0.3834
I=21    -0.3409   -0.2411    0.0       0.2411    0.3409
I=22    -0.2671   -0.1888    0.0       0.1888    0.2671
I=23    -0.1568   -0.1108    0.0       0.1108    0.1568
I=24     0.0       0.0       0.0       0.0       0.0
```

NETWORK POINTS ON UPPER FIN

```
                                IW=2          J=1 IS TIP
                                              J=NJ IS ROOT
XNET(I,J,IW)
I= 1    -3.9450   -3.8175   -3.6900   -3.5625   -3.4350
I= 2    -4.0012   -3.8897   -3.7781   -3.6666   -3.5550
I= 3    -4.0575   -3.9619   -3.8642   -3.7706   -3.6750
I= 4    -4.1137   -4.0341   -3.9544   -3.8747   -3.7950
I= 5    -4.1700   -4.1062   -4.0425   -3.9787   -3.9150
I= 6    -4.2242   -4.1784   -4.1306   -4.0828   -4.0350
I= 7    -4.2825   -4.2506   -4.2187   -4.1869   -4.1550
I= 8    -4.3387   -4.3225   -4.3069   -4.2909   -4.2750
I= 9    -4.3950   -4.3950   -4.3950   -4.3950   -4.3950

YNET(I,J,IW)
I= 1    -0.4020   -0.3513   -0.3005   -0.2498   -0.1991
I= 2    -0.4020   -0.3513   -0.3005   -0.2498   -0.1991
I= 3    -0.4020   -0.3513   -0.3005   -0.2495   -0.1991
I= 4    -0.4020   -0.3513   -0.3005   -0.2498   -0.1991
I= 5    -0.4020   -0.3513   -0.3005   -0.2498   -0.1991
I= 6    -0.4020   -0.3513   -0.3005   -0.2498   -0.1991
I= 7    -0.4020   -0.3513   -0.3005   -0.2498   -0.1991
I= 8    -0.4020   -0.3513   -0.3005   -0.2498   -0.1991
I= 9    -0.4020   -0.3513   -0.3005   -0.2498   -0.1991

ZNET(I,J,IW)
I= 1    -0.4020   -0.3513   -0.3005   -0.2498   -0.1991
I= 2    -0.4020   -0.3513   -0.3005   -0.2498   -0.1991
I= 3    -0.4020   -0.3513   -0.3005   -0.2498   -0.1991
I= 4    -0.4020   -0.3513   -0.3005   -0.2495   -0.1991
I= 5    -0.4020   -0.3513   -0.3005   -0.2498   -0.1991
I= 6    -0.4020   -0.3513   -0.3005   -0.2498   -0.1991
I= 7    -0.4020   -0.3513   -0.3005   -0.2498   -0.1991
I= 8    -0.4020   -0.3513   -0.3005   -0.2498   -0.1991
I= 9    -0.4020   -0.3513   -0.3005   -0.2498   -0.1991
```

NETWORK POINTS ON LOWER FIN

```
                                IW=3          J=1 IS TIP
                                              J=NJ IS ROOT
XNET(I,J,IW)
I= 1    -3.9450   -3.5178   -3.6900   -3.5625   -3.4350
I= 2    -4.0012   -3.8897   -3.7781   -3.6666   -3.5550
I= 3    -4.0575   -3.9619   -3.8662   -3.7706   -3.6750
I= 4    -4.1137   -4.0341   -3.9544   -3.8747   -3.7950
I= 5    -4.1700   -4.1062   -4.0425   -3.9787   -3.9150
I= 6    -4.2262   -4.1784   -4.1306   -4.0825   -4.0350
I= 7    -4.2825   -4.2506   -4.2187   -4.1569   -4.1550
I= 8    -4.3387   -4.3225   -4.3069   -4.2909   -4.2750
I= 9    -4.3950   -4.3950   -4.3950   -4.3950   -4.3950

YNET(I,J,IW)
I= 1    -0.4020   -0.3513   -0.3005   -0.2498   -0.1991
I= 2    -0.4020   -0.3513   -0.3005   -0.2498   -0.1991
I= 3    -0.4020   -0.3513   -0.3005   -0.2498   -0.1991
I= 4    -0.4020   -0.3513   -0.3005   -0.2498   -0.1991
I= 5    -0.4020   -0.3513   -0.3005   -0.2498   -0.1991
I= 6    -0.4020   -0.3513   -0.3005   -0.2498   -0.1991
I= 7    -0.4020   -0.3513   -0.3005   -0.2498   -0.1991
I= 8    -0.4020   -0.3513   -0.3005   -0.2498   -0.1991
I= 9    -0.4070   -0.3513   -0.3005   -0.2498   -0.1991

ZNET(I,J,IW)
I= 1     0.4020    0.3513    0.3005    0.2498    0.1991
I= 2     0.4020    0.3513    0.3005    0.2498    0.1991
I= 3     0.4020    0.3513    0.3005    0.2498    0.1991
I= 4     0.4020    0.3513    0.3005    0.2498    0.1991
I= 5     0.4020    0.3513    0.3005    0.2498    0.1991
I= 6     0.4020    0.3513    0.3005    0.2495    0.1991
I= 7     0.4020    0.3513    0.3005    0.2498    0.1991
I= 8     0.4020    0.3513    0.3005    0.2498    0.1991
I= 9     0.4020    0.3513    0.3005    0.2498    0.1991
```

Output Data

The output data, which is passed to the following job step using data set reference number 51, consists of the inputs to this step plus arrays that are functions of the geometry and vortex modeling.

2.5 USER'S GUIDE TO PROGRAM C–THIRD JOB STEP

Input Data

The following are input data for the optional version of the third job step of Program C, in which trajectories are computed.

LM — Object-time dimension of the dependent variables of the equations of motion

AMACH — Free-stream Mach number

DT(1) — Time interval of half of a numerical integration step, sec

NCASE — Number of vortex-lattice cases to be computed with each transferal of data from data set reference number 21 into the computer core

NK — Number of time steps plus 1 in the trajectory; NK-1 must be a multiple of NCASE-1

SCALE — Ratio of full-scale store dimensions to store model dimensions

WGTTA — Components of full-scale store weight along X, Y, and Z wind axes, lb

S — Full-scale store reference area, ft^2

BSPAN — Full-scale store lateral reference length, ft

CBAR — Full-scale store longitudinal reference length, ft

AIX, AIY, AIZ — Full-scale store mass moments of inertia about body axes, slug-ft^2

AIXZ — Full-scale store cross product of mass moment of inertia in the body x-z plane, slug-ft^2

RHO — Free-stream density, slug/ft^3

ASOUND — Speed of sound, ft/sec

FX1, FX2 Thrust forces acting along full-scale store body x-axis (see Fig. 12, Ref. 4), lb

FZ1, FZ2 Ejector forces acting on full-scale store (see Fig. 12, Ref. 4), lb

ALX1, ALX2 Body x-axis coordinates of full-scale store ejection stations (see Fig. 12, Ref. 4), ft

CLP Roll-damping coefficient (see p. viii of Ref. 4)

CMQ Pitch-damping coefficient (see p. viii of Ref. 4)

CNR Yaw-damping coefficient (see p. viii of Ref. 4)

U(1), V(1), W(1) Initial values of full-scale store translational velocity components along body x axis, y axis, and z axis, respectively, ft/sec

P(1), Q(1), R(1) Initial values of full-scale store rotational velocity components about body x axis, y axis, and z axis, respectively, radians/sec

X(1), Y(1), Z(1) Initial values of full-scale store center of gravity coordinates in wind axes, ft

ANU(1), PSI(1), OMEGA(1) Initial values of angular displacement about the store body axes in pitch, yaw, and roll, respectively, radians/sec

CF(1, 1), . . . , CF(1, 6) Initial values of aerodynamic force coefficients in the sequence axial force, side force, normal force, pitching moment, yawing moment, and rolling moment

CA1 Coefficient used to correct the axial-force coefficient in the predictor-corrector numerical integration scheme

CLA Coefficient used to correct the normal- and side-force coefficients in the predictor-corrector numerical integration scheme

CMA Coefficient used to correct the pitching- and yawing-moment coefficients in the predictor-corrector numerical integration scheme

The following are input data for the optional version of the third job step of Program C, in which force coefficients, but not trajectories, are computed.

NCASE Number of vortex-lattice cases to be computed with each transferal of data from data set reference Number 21 into the core

NREAD Number of times data set reference Number 21 is read

X, Y, Z Coordinates of the point which the store body axes are rotated about with respect to the wind axes, in the parent aircraft reference system, ft

ANU, PSI, OMEGA Angular displacement about the store body axes in pitch, yaw, and roll, respectively, radians/sec

The following input data are defined by arithmetic expressions:

ICF = 1 Laminar skin friction assumed
= 2 Turbulent skin friction assumed

BSPAN Full-scale store lateral reference length, ft

CBAR Full-scale store longitudinal reference length, ft

Dimensions of Arrays

Whenever the number 195 appears as a dimension in the program listing, this is the number of points at which velocity is calculated when ISOLVE = 3 (NUT(3)). The arithmetic expression for NUT(3) is given in the second job step of Program C.

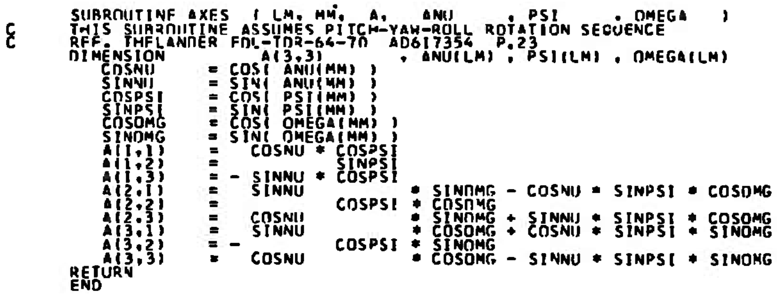
NCASE (which has the value 5 in the sample program listing) is a dimension for the following variables: CDTOT, CYTOT, CLTOT, CMTOT, CNTOT, CRTOT, BDOTV, GAMMAV, HGAMMA, VX, VY, and VZ.

In SUBROUTINE ACOEFF the elements of the vorticity distribution in the GAMMAV array are reordered to form the GAMMA array. The maximum values of the three subscripts of GAMMA are, respectively, (1) the maximum value of NCHORD(IW), (2) the maximum value of NSPAN(IW) + 1, and (3) NWP. In this same subroutine the dimensions of the variables which are the spanwise distributions of force coefficients on vortex segments are, respectively, (1) the maximum value of NCHORD(IW) and (2) the maximum value of NSPAN(IW)-1. The dimensions of the variables which represent chordwise distributions are, respectively, (1) the maximum value of NCHORD(IW)-1 and (2) the maximum value of NSPAN(IW).

```
C     PROGRAM C  -  3RD JOB STEP
C     THIS PROGRAM INTEGRATES THE SIX-DEGREE-OF-FREEDOM
C     EQUATIONS OF MOTION USING A 4TH ORDER RUNGE-KUTTA METHOD
      DIMENSION
     1   L(4)           , DT     (33)    ,
     2   A(3,3)         ,
     3   WGTTA(3)       , WGTBA(3)       ,
     4   CF      (33,6) , DCFDA(6,2)     ,
     4   PITCH (33)     , YAW    (33)    , PITCHP(33)    , YAWP  (33)    ,
     4   AINCID(33)     , AINCP  (33)    ,
     5   CA(33)     ,CY(33)     ,CN(33)     ,CPITCH(33),CYAW(33),CROLL(33),
     6   CDTOT(5)   ,CYTOT(5)   ,CLTOT(5)   ,CMTOT(5)  ,CNTOT(5)  ,CRTOT(5)  ,
     8   UDOT(4)    ,VDOT(4)    ,WDOT(4)    ,PDOT(4)   ,QDOT(4)   ,RDOT(4)   ,
     9   XDOT(4)    ,YDOT(4)    ,ZDOT(4)    ,ANUDOT(4),PSIDOT(4),OMGDOT(4),
     A   U(33)      ,V(33)      ,W(33)      ,P(33)     ,Q(33)     ,R(33)     ,
     B   X(33)      ,Y(33)      ,Z(33)      ,ANU(33)   ,PSI(33)   ,OMEGA(33)
      CALL ERRSET (209,10,5,2)                                             DIV CHK
      CALL ERRSET (251,10,5,2)                                             NEG SQRT
      CALL ERRSET (252,10,5,2)                                             BIG EXPN
      CALL ERRSET (253,10,5,2)                                             NEG LG10
C     ALTITUDE            = 5000.
C     TEMP                =   41.2 DEG F
C     ASOUND              = 1097.5
C     EJECTOR FORCE       = 1000.
C     EJECTOR STROKE =          .2552
C     XCG FULL SCALE =        2.74
C
C     INPUTS SUBJECT TO FREQUENT CHANGE
C     U(1),....,OMEGA(1), CF(1,JF)
      LN        =    33
      NK        =    17
      NM        = 2*NK - 1
      NCASE     =     5
      INCREM    = (NCASE-1) / 2
      AMACH     =     0
      DT(1)     = .1 / 4.
      DO 500 M=1,NM
  500 DT(M)     = DT(1)
C
C
C
C     THE FOLLOWING STATEMENTS ARE PECULIAR TO THE M-117 BOMB
      SCALE     =   20.
      WGTTA(1)=       0.
      WGTTA(2)=       0.
      WGTTA(3)=   250.
      AMBAR     = WGTTA(3) / 32.2
      S         =     1.395
      BSPAN     =     1.333
      CBAR      =     7.325
      AIX       =     4.
      AIY       =    30.
      AIZ       =    30.
      AIXZ      =      0.
C
      RHO       =       .001978
      ASOUND    = 1097.5
      U1        = AMACH * ASOUND
C
C     THRUST AND EJECTOR FORCES
      FX1       =     0.
      FX2       =     0.
      FZ1       =     0.
      FZ2       =     0.
      ALX1      =     0.
      ALX2      =     0.
C
C     DAMPING COEFFICIENTS
      CLP       =     0.
      CMQ       =    -2.319
      CNR       =    -2.319
C
      XCARIG    = SCALE * (-12.99-1.644) / 12.
      YCARIG    = SCALE * (-4.66)        / 12.
      ZCARIG    = SCALE * 1.1 / 12.
      GO TO 600
  600 CONTINUE
C     INITIAL CONDITIONS FOR BOMB RELEASE
      P(1)      =     0.
      Q(1)      =     0.
      R(1)      =     0.
C     IF OMEGA(1)=3.1416/4. BOTH ANU(1) AND PSI(1) .NE. 0.
      ANU(1)    =     -.7 * 3.1416 / 180.
      PSI(1)    =     0.
      OMEGA(1)=       3.1416 / 4.
      OMEGA(1)=       0.
      GO TO 620
C
C     VERTICAL EJECTION OF BOMB
      U(1)      =     0.
      V(1)      =     0.
      W(1)      =     9.
      X(1)      = XCARIG
      Y(1)      = YCARIG
      Z(1)      = ZCARIG + .2552
      GO TO 650
C
  620 CONTINUE
C     45 DEGREE EJECTION OF BOMB
      U(1)      =     0.
      V(1)      =    -8.79 / SQRT( 2. )
      W(1)      =     8.79 / SQRT( 2. )
      X(1)      = XCARIG
      Y(1)      = YCARIG  -  .2552 / SQRT( 2. )
      Z(1)      = ZCARIG  +  .2552 / SQRT( 2. )
      CF(1,1) =   .1346
      CF(1,2) =   .0485
      CF(1,3) =   .0032
      CF(1,4) = -.0008
      CF(1,5) = -.0128
      CF(1,6) =   .0
      GO TO 650
C
C     INITIAL CONDITIONS FOR CONTINUATION OF TRAJECTORY CALCULATIONS
C     U(1),....,OMEGA(1), CF(1,JF)
C
C
  650 CONTINUE
C     STATIC STABILITY DERIVATIVES
C     CLA             = D(CL)/D(ANU)
```

```
C      CMA            = D(CM)/D(ANU)
C      CYP            = D(CY)/D(PSI)
C      CNP            = D(CN)/D(PSI)
       IF( ABS(AMACH-0.5 )  -  .1 )  820, 820, 810
  810  IF( ABS(AMACH-0.85)  -  .1 )  830, 830, 840
  820  CLA            =   .0636 * 180. / 3.1416
       CMA            = -.0169 * 180. / 3.1416
       CA1            = ( (.134-.064)/10.**2 )   *  (180./3.1416)**2
       GO TO 850
C
  830  CLA            =  (.39/5.7) * 180. / 3.1416
       CMA            = -(.1/5.7) * 180. / 3.1416
       CA1            = ( (.089-.066)/5.7**2 )   *  (180./3.1416)**2
       GO TO 850
C
  840  WRITE(6,8122)
 8122  FORMAT(     ' MACH NO. NOT PROPFRLY INPUT CAUSING PROGRAMMED STOP')
       STOP
C      THE ABOVE STATEMENTS ARE PECULIAR TO THE M-117 BOMB
C
C
C
C
  850  CYP            = CLA
       CNP            = CMA
C      SIDE FORCF
       DCFDA(2,1) = 0.
       DCFDA(2,2) = CYP
C      LIFT
       DCFDA(3,1) = CLA
       DCFDA(3,2) = 0.
C      PITCH
       DCFDA(4,1) = CMA
       DCFDA(4,2) = 0.
C      YAW
       DCFDA(5,1) = 0.
       DCFDA(5,2) = CNP
C      ROLL
       DCFDA(6,1) = 0.
       DCFDA(6,2) = 0.
C
       IKJ  = 0
       L(1) = 0
       L(2) = 1
       L(3) = 1
       L(4) = 2
C
       DO 6300    KM1=1,NK, INCREM
       M1                = 2* KM1           - 1
       M2                = 2*(KM1+INCREM) - 1
       DO 6200   ITCF=1,2
       DO 6100    KM2=1,INCREM
       K    = KM1 + KM2 - 1
       M    = 2*K - 1
          GO TO( 2500, 2210), ITCF
 2210     IF( KM2-1 )  2220, 2220, 2500
C
 2220     CALL     VORLAT( LM, K, M,
     A    NCASE           ,
     1    IKJ             ,
     1    U1              , SCALE       ,
     A    U        , V          , W          , P        , Q        , R        ,
     B    X        , Y          , Z          , ANU      , PSI      , OMEGA    ,
     C    CDTOT    , CYTOT      , CLTOT      , CMTOT    , CNTOT    , CRTOT    )
C
          DO 2410 ICASE=1,NCASE
          MM           = M-1+ICASE
          CF(MM,1)     = CDTOT(ICASE)
          CF(MM,2)     = CYTOT(ICASE)
          CF(MM,3)     = CLTOT(ICASE)
          CF(MM,4)     = CMTOT(ICASE)
          CF(MM,5)     = CNTOT(ICASE)
          CF(MM,6)     = CRTOT(ICASE)
 2410     CONTINUE
C
          WRITE(6,8220)
 8220     FORMAT(//      ' 2ND ESTIMATE OF AERODYNAMIC COEFFICIENTS FROM',
     1    ' VORTEX-LATTICE CALCULATIONS '        /  13X,
     2       ' CAXIAL      CSIDE        CNORM        CPITCH      CYAW        CROLL ')
C
          DO 8230 MM=M1,M2
 8230     WRITE(6,8232)   MM,   (CF(MM,JF), JF=1,6)
 8232     FORMAT(   '      M=',I2, 2X,   6F10.4   )
C
 2500     CONTINUE
       KRK   = 0
 3000  KRK   = KRK + 1
       M     = 2*K - 1 + L(KRK)
       IF( K-NK   )   3110, 6500, 6500
 3110  IF( KRK-1 )   3500, 3500, 3120
 3120     CONTINUE
          U(M)           = U(2*K-1)   +   L(KRK) * UDOT(KRK-1) * DT(M-1)
          V(M)           = V(2*K-1)   +   L(KRK) * VDOT(KRK-1( * DT(M-1)
          W(M)           = W(2*K-1)   +   L(KRK) * WDOT(KRK-1) * DT(M-1)
          P(M)           = P(2*K-1)   +   L(KRK) * PDOT(KRK-1) * DT(M-1)
          Q(M)           = Q(2*K-1)   +   L(KRK) * QDOT(KRK-1) * DT(M-1)
          R(M)           = R(2*K-1)   +   L(KRK) * RDOT(KRK-1) * DT(M-1(
          X(M)           = X(2*K-1)   +   L(KRK) * XDOT(KRK-1) * DT(M-1)
          Y(M)           = Y(2*K-1)   +   L(KRK) * YDOT(KRK-1) * DT(M-1)
          Z(M)           = Z(2*K-1)   +   L(KRK) * ZDOT(KRK-1) * DT(M-1)
          ANU(M)         = ANU(2*K-1)   +   L(KRK) * ANUDOT(KRK-1) * DT(M-1)
          PSI(M)         = PS((2*K-1)   +   L(KRK) * PSIDOT(KRK-1) * DT(M-1)
          OMEGA(M)= OMEGA(2*K-1)   +   L(KRK) * OMGDOT(KRK-1) * DT(M-1)
 3500     CONTINUE
C
C
          PITCH(M)   = ANU(M)   +   ATAN2( W(M),U1+U(M) )
          YAW(M)     = PSI(M)   -   ATAN2( V(M),U1+U(M) )
          AINCID(M) = ARCOS(   COS(PITCH(M)) * COS(YAW(M))   )
          GO TO( 4100, 4200), ITCF
 4100     ITCF          = 1
          M1            = 2*KM1 - 1
          CF(M,1)       = CF(M1,1)     -  CA1 * (AINCID(M)**2 - AINCID(M1)**2)
          DO 4150 JF=2,6
 4150     CF(M,JF)      = CF(M1,JF)    +  DCFDA(JF,1) * (PITCH(M)-PITCH(M1))
     1                                 +  DCFDA(JF,2) * (YAW(M)   -YAW(M1)   )
          GO TO 4300
C
 4200     ITCF          = 2
          CF(M,1)       = CF(M,1)      -  CA1 * (AINCID(M)**2 - AINCP(M)**2)
          DO 4250 JF=2,6
 4250     CF(M,JF)   = CF(M,JF)    +  DCFDA(JF,1) * (PITCH(M)-PITCMP(M))
      1                            +  DCFDA(JF,2) * (YAW(M)  -YAWP(M(  )
 4300     CONTINUE
          CA(M)      = CF(M,1)
          CY(M)      = CF(M,2)
          CN(M)      = CF(M,3)
          CPITCH(M)  = CF(M,4)
          CYAW(M)    = CF(M,5)
          CROLL(M)   = CF(M,6)
C
C
C
          CALL    AXES  ( LM, M,   A,    ANU   ,    PSI   ,    OMEGA )
          DO 4500  N=1,3
 4500     WGTBA(N)   = A(N,1)*WGTTA(1) + A(N,2)*WGTTA(2) + A(N,3)*WGTTA(3)
C
C         THE FORMULAS FOR THE DERIVATIVES XDOT(KRK),....,RDOT(KRK)
C         ARE GIVEN IN AEDC-TR-68-200, BY CHRISTOPHER AND CARLTON
          XDOT(KRK) = A(1,1) * U(M)   +   A(2,1) * V(M)   +   A(3,1) * W(M)
          YDOT(KRK) = A(1,2) * U(M)   +   A(2,2) * V(M)   +   A(3,2) * W(M)
          ZDOT(KRK) = A(1,3) * U(M)   +   A(2,3) * V(M)   +   A(3,3) * W(M)
          COSPSI     = COS( PSI(M) )
          SINPSI     = SIN( PSI(M) )
          COSOMG     = COS( OMEGA(M) )
          SINOMG     = SIN( OMEGA(M) )
          ANUDOT(KRK) = ( Q(M) * COSOMG - R(M) * SINOMG )  /  COSPSI
          PSIDOT(KRK) =    R(M) * COSOMG + Q(M) * SINOMG
          OMGDOT(KRK) =    P(M) - ANUDOT(KRK) * SINPSI
C
          UTOTAL     = SQRT( (U1+XDOT(KRK))**2 + YDOT(KRK)**2
      1                                        + ZDOT(KRK)**2  )
          QS         = RHO * UTOTAL**2 / 2.
          FX         = WGTBA(1)  -  QS * S * CA(M)   -  FX1  +  FXZ
          FY         = WGTBA(2)  +  QS * S * CY(M)
          FZ         = WGTBA(3)  -  QS * S * CN(M)   +  FZ1  +  FZ2
          TL         = CROLL(M)    +  CLP * BSPAN* P(M) / (2.*UTOTAL)
          TM         = CPITCH(M)   +  CMQ * CBAR * Q(M) / (2.*UTOTAL)
          TN         = CYAW(M)     +  CNR * CBAR * R(M) / (2.*UTOTAL)
          TL         = QS * S * BSPAN* TL
          TM         = QS * S * CBAR * TM   -  FZ1 * ALX1   -  FZ2 * ALX2
          TN         = QS * S * CBAR * TN
          UDOT(KRK) = FX / AMBAR  +  R(M) * V(M)  -  Q(M) * W(M)
          VDOT(KRK) = FY / AMBAR  +  P(M) * W(M)  -  R(M) * U(M)
          WDOT(KRK) = FZ / AMBAR  +  Q(M) * U(M)  -  P(M) * V(M)
          PNDOT      = TL  -  Q(M)*R(M)*( AIZ-AIY )  +  P(M)*Q(M)*AIXZ
          PNDOT      = PNDOT / AIX
          RNDOT      = TN  -  Q(M)*R(M)*AIXZ  -  P(M)*Q(M1*( AIY-AIX )
          RNDOT      = RNDOT / AIZ
          PDOT(KRK) = ( PNDOT  +  RNDOT * AIXZ / AIX )
      1                      / ( 1. - AIXZ**2 / (AIX*AIZ) )
          QDOT(KRK) = TM  -  ( P(M)**2-R(M)**2 ) * AIXZ
      1                   -  P(M) * R(M) * ( AIX-AIZ )
          QDOT(KRK) = QDOT(KRK) / AIY
          RDOT(KRK) = TN  +  ( PDOT(KRK)-Q(M)*R(M) ) * AIXZ
      1                   -  P(M) * Q(M) * ( AIY-AIX )
          RDOT(KRK) = RDOT(KRK) / AIZ
       GO TO( 3000, 3000, 3000, 5100), KRK
 5100     CONTINUE
          U(M)       = U(2*K-1) + (UDOT(1)+2.*UDOT(2)+2.*UDOT(3)+UDOT(4))
      1                         * DT(M-1) / 3.
          V(M)       = V(2*K-1) + (VDOT(1)+2.*VDOT(2)+2.*VDOT(3)+VDOT(4))
      1                         * DT(M-1) / 3.
          W(M)       = W(2*K-1) + (WDOT(1)+2.*WDOT(2)+2.*WDOT(3)+WDOT(4))
      1                         * DT(M-1) / 3.
          P(M)       = P(2*K-1) + (PDOT(1)+2.*PDOT(2)+2.*PDOT(3)+PDOT(4))
      1                         * DT(M-1( / 3.
          Q(M)       = Q(2*K-1) + (QDOT(1)+2.*QDOT(2)+2.*QDOT(3)+QDOT(4()
      1                         * DT(M-1) / 3.
          R(M)       = R(2*K-1) + (RDOT(1)+2.*RDOT(2)+2.*RDOT(3)+RDOT(4))
      1                         * DT(M-1) / 3.
          X(M)       = X(2*K-1) + (XDOT(1)+2.*XDOT(2)+2.*XDOT(3)+XDOT(4))
      1                         * DT(M-1) / 3.
          Y(M)       = Y(2*K-1) + (YDOT(1)+2.*YDOT(2)+2.*YDOT(3)+YDOT(4))
      1                         * DT(M-1) / 3.
          Z(M)       = Z(2*K-1) + (ZDOT(1)+2.*ZDOT(2)+2.*ZDOT(3)+ZDOT(4))
      1                         * DT(M-1) / 3.
          ANU(M)     = ANU(2*K-1) + ( ANUDOT(1)+2.*ANUDOT(2(
      1                         +2.*ANUDOT(3)+ANUDOT(4(  ) *DT(M-1)/3.
          PSI(M)     = PSI(2*K-1) + ( PSIDOT(1)+2.*PSIDOT(2)
      1                         +2.*PSIDOT(3)+PSIDOT(4(  ) *DT(M-1)/3.
          OMEGA(M)= OMEGA(2*K-1) + ( OMGDOT(1)+2.*OMGDOT(2)
      1                         +2.*OMGDOT(3)+OMGDOT(4)  ) *DT(M-1)/3.
C
          IF( ITCF-1 )  5510, 5510, 6100
 5510     IF( M-M2   )  6100, 5520, 5520
 5520     DO 5550  M=M1,M2
          PITCHP(M) = PITCH(M)
          YAWP(M)    = YAW(M)
          AINCP(M)   = AINCIO(M)
 5550     CONTINUE
 6100 CONTINUE
      IF( KM1-(NK-INCREM) )  8622, 8624, 6200
 8622 MI = M1
      GO TO 8630
 8624 MI = 1
 8630 WRITE(6,8632)
 8632 FORMAT(// ' AERODYNAMIC COEFFICIENTS FROM TAYLOR SERIES' /  3X,
     1 ' ITCF M     CAXIAL     CSIDE        CNORM       CPITCH      CYAW        CRO
     2LL    ')
      DD 8640        M=MI,M2
 8640 WRITE(6,8642) ITCF, M,  CA(M), CY(M), CN(M),
     1                        CPITCH(M), CYAW(M), CROLL(M)
 8642 FORMAT(  2I5,  6F10.4     )
      WRITE(6,8646)
 8646 FORMAT(//  ' U,V,W,P,Q,R,ANU,PSI,OMEGA  ARE REFERRED TO BODY AXES.
     1  UNITS ARE FEET, SECONDS, DEGREES.  ')
      WRITE(6,8652)
 8652 FORMAT(       3X,   ' ITCF M ',
     1  3X, ' U(M)         V(M)         W(M)         P(M)         Q(M)        R(M)   ',
     2  4X, ' X(M)         Y(M)         Z(M)         ANU(M)       PSI(M)      OMEGA(M)'
     3  )
      DO 8660        M=MI,M2
      PDEG           = P(M)        * 180. / 3.1416
      QDEG           = Q(M)        * 180. / 3.1416
      RDEG           = R(M)        * 180. / 3.1416
      ANUDEG         = ANU(M)      * 180. / 3.1416
      PSIDEG         = PSI(M)      * 180. / 3.1416
      OMGDEG         = OMEGA(M)    * 180. / 3.1416
 8660 WRITE(6,8662)  ITCF, M, U(M), V(M), W(M), PDEG, QDEG, RDEG,
     1                        X(M), Y(M), Z(M), ANUDEG, PSIDEG, OMGDEG
 8662 FDRMAT(2I5,     12F10.3  )
C
 6200 CONTINUE
 6300 CONTINUE
 6500 CONTINUE
C
C
      WRITE(6,8720)
 8720 FORMAT( ///   ' TRAJECTORY VELDCITY AND DISPLACEMENT IN WIND AXES '
     1          /   ' UNITS ARE FEET, SECDNDS, DEG '
     2          /   '      M', 3X, ' U(FT/SEC) V(FT/SEC) W(FT/SEC) P(DEG/S)
     3  Q(DEG/S)  R(DEG/S)  X(FT)      Y(FT)      Z(FT)      ANU(DEG)  PSI(
     4DEG)  OMEGA(DEG)   ')
      DO 6600   M=1,NM
      CALL        AXES   ( LM, M,    A,    ANU    ,    PSI    ,    OMEGA )
      UWIND    = A(1,1) * U(M)   +   A(2,1) * V(M)   +   A(3,1) * W(M)
      VWIND    = A(1,2) * U(M)   +   A(2,2) * V(M)   +   A(3,2) * W(M)
      WWIND    = A(1,3) * U(M)   +   A(2,3) * W(M)   +   A(3,3) * W(M)
C
      PWIND    = A(1,1) * P(M)   +   A(2,1) * Q(M)   +   A(3,1) * R(M)
      QWIND    = A(1,2) * P(M)   +   A(2,2) * Q(M)   +   A(3,2) * R(M)
      RWIND    = A(1,3) * P(M)   +   A(2,3) * Q(M)   +   A(3,3) * R(M)
C
      OMEGAW   =A(1,1) * OMEGA(M)   +   A(2,1) * ANU(M)   +   A(3,1) * PSI(M)
      ANUW     =A(1,2) * OMEGA(M)   +   A(2,2) * ANU(M)   +   A(3,2) * PSI(M)
      PSIW     =A(1,3) * OMEGA(M)   +   A(2,3) * ANU(M)   +   A(3,3) * PSI(M)
C
      PWIND    = PWIND * 180. / 3.1416
      QWIND    = QWIND * 180. / 3.1416
      RWIND    = RWIND * 180. / 3.1416
      ANUW     = ANUW  * 180. / 3.1416
      PSIW     = PSIW  * 180. / 3.1416
      OMEGAW   = OMEGAW* 180. / 3.1416
      WRITE(6,8730)       M, UWIND, VWIND, WWIND, PWIND, QWIND, RWIND,
     1                      X(M), Y(M), Z(M), ANUW, PSIW, OMEGAW
 8730 FORMAT(      I5,    12F10.3 )
 6600 CONTINUE
      STOP
      END
```

```
C     PROGRAM C  -  3RD JOB STEP
C     THIS IS AN ALTERNATE VERSION OF THE MAIN PROGRAM OF THIS STEP IN
C     WHICH TRAJECTORIES ARE NOT COMPUTED.
C     THIS PROGRAM COMPUTES AERODYNAMIC FORCES ON A STATIONARY PLANFORM
C     WHOSE ORIENTATION IN SPACE IS SPECIFIED
      DIMENSION
     1   U     (5), V     (5), W     (5), P     (5), Q     (5), R     (5),
     2   X     (5), Y     (5), Z     (5), ANU   (5), PSI   (5), OMEGA(5),
     3   CDTOT(5), CYTOT(5), CLTOT(5), CMTOT(5), CNTOT(5), CRTOT(5)
      DIMENSION              A(3,3)
 7100 FORMAT( 16I5    )
 7200 FORMAT( 8E10.0 )
      LM          = 5
      K           = 1
      M           = 1
C     INPUT DUMMY VALUES FOR U1 AND SCALE
      U1          = 1.
      SCALE       = 1.
      READ(5,7100)          NCASE,     NREAD
      DO 100 MM=1,NCASE
      U(MM)       = 0.
      V(MM)       = 0.
      W(MM)       = 0.
      P(MM)       = 0.
      Q(MM)       = 0.
  100 R(MM)       = 0.
C
      IKJ         = 0
C
C
      DO 200    IREAD=1,NREAD
      READ(5,7200)        (X(MM),      MM=1,NCASE)
      READ(5,7200)        (Y(MM),      MM=1,NCASE)
      READ(5,7200)        (Z(MM),      MM=1,NCASE)
      READ(5,7200)        (ANU(MM),    MM=1,NCASE)
      READ(5,7200)        (PSI(MM),    MM=1,NCASE)
      READ(5,7200)        (OMEGA(MM),  MM=1,NCASE)
      CALL           VORLAT( LM, K, M,
     1    NCASE           ,
     1    IKJ             ,
     1   ,U1              , SCALE          ,
     A    U         , V         , W         , P         , Q         , R         ,
     B    X         , Y         , Z         , ANU       , PSI       , OMEGA     ,
     C    CDTOT     , CYTOT     , CLTOT     , CMTOT     , CNTOT     , CRTOT     )
      ICF         = 1
      RSPAN       =   .8   / 12.
      CRAR        = 4.395  / 12.
      DO 8900     MM=1,NCASE
      XORIG       = X(MM) * 12.
      YORIG       = Y(MM) * 12.
      ZORIG       = Z(MM) * 12.
      PITCH       = ANU(MM) * 180. / 3.1416
      YAW         = PSI(MM) * 180. / 3.1416
      ROLL        = OMEGA(MM) * 180. / 3.1416
      CALL          AXES  ( LM, MM, A,    ANU    ,  PSI    ,  OMEGA )
      CD          =        A(1,1) * CDTOT(MM)  -  A(2,1) * CYTOT(MM)
     1                  +  A(3,1) * CLTOT(MM)
      CY          = -      A(1,2) * CDTOT(MM)  +  A(2,2) * CYTOT(MM)
     1                  -  A(3,2) * CLTOT(MM)
      CL          =        A(1,3) * CDTOT(MM)  -  A(2,3) * CYTOT(MM)
     1                  +  A(3,3) * CLTOT(MM)
      CR          =        A(1,1)*RSPAN*CRTOT(MM)  +  A(2,1)*CRAR*CMTOT(MM)
     1                  +  A(3,1)*CRAR*CNTOT(MM)
      CM          =        A(1,2)*RSPAN*CRTOT(MM)  +  A(2,2)*CRAR*CMTOT(MM)
     1                  +  A(3,2)*CRAR*CNTOT(MM)
      CN          =        A(1,3)*RSPAN*CRTOT(MM)  +  A(2,3)*CRAR*CMTOT(MM)
     1                  +  A(3,3)*CRAR*CNTOT(MM)
      CR          = CR / RSPAN
      CM          = CM / CRAR
      CN          = CN / CRAR
C
      WRITE(6,8720)  MM,       XORIG, YORIG, ZORIG, PITCH, YAW, ROLL
 8720 FORMAT(//     ' IRUN=',I2, 2X, ' XORIG(IN)=',F6.2, 2X,
     1 ' YORIG(IN)=',F6.2, 2X, ' ZORIG(IN)=',F6.2, 2X,
     2 ' PITCH(DEG)=',F6.1, 2X, ' YAW(DEG)=',F6.1, 2X, ' ROLL(DEG)=',
     3 F6.1   )
      GO TO( 8820, 8830), ICF
 8820 WRITE(6,8822)
 8822 FORMAT(  15X,     ' FORCE COEFFICIENTS FROM  LAMINAR  SKIN FRICTION
     1ASSUMPTION ')
      GO TO 8840
 8830 WRITE(6,8832)
 8832 FORMAT(/ 15X,     ' FORCE COEFFICIENTS FROM TURBULENT SKIN FRICTION
     1ASSUMPTION ')
 8840 WRITE(6,8842)     MM, CD, CY, CL, CR, CM, CN
 8842 FORMAT(   15X,    ' FORCE COEFFICIENTS IN WIND AXES SYSTEM '
     1         / 15X,   ' CD(DRAG)           CY(SIDE FORCE) CL(LIFT)          CR(
     2ROLL)           CM(PITCH)          CN(YAW)    ' / ' IRUN=',I2,   6F15.5    )
      WRITE(6,8852)  MM, CDTOT(MM), CYTOT(MM), CLTOT(MM),
     1                   CRTOT(MM), CMTOT(MM), CNTOT(MM)
 8852 FORMAT(   15X,    ' FORCE COEFFICIENTS IN BODY AXES SYSTEM '
     1         / 15X,   ' CAXIAL            CSIDE             CNORM             CRO
     2LL           CPITCH            CYAW     ' / ' IRUN=',I2,   6F15.5    (
 8900 CONTINUE
  200 CONTINUE
      STOP
      END
      SUBROUTINE VORLAT( LM, K, M,
     A    NCASE          ,
     1    IKJ            ,
     1    U1             , SCALE          ,
     A    UCG       , VCG       , WCG       , P         , Q         , R         ,
     B    X         , Y         , Z         , ANU       , PSI       , OMEGA     ,
     C    CDTOT     , CYTOT     , CLTOT     , CMTOT     , CNTOT     , CRTOT     )
      COMMON/CNET  /      NWP             , NIWP            ,
     1    NCHORD(3)     , NSPAN (3)       , NSCORD(3)       , NVOR   (3)      ,
     2    NUC    (3,4)  , NUS    (3,4)    , NVEL   (3,4)    , NUT       (4)
      COMMON/CCOORD/      XNET (24,5,3), YNET (24,5,3), ZNET (24,5,3),
     1                    AKSI (23,4,3), ETA  (23,4,3), ZETA (23,4,3),
     2                    XFLOWF( 1, 1), YFLOWF( 1, 1), ZFLOWF( 1, 1),
     3    BDC    (156,3), AX          (3), AY          (3), AZ          (3)
      COMMON/CVELOC/      U     (24,5,3), V     (24,5,3), W     (24,5,3),
     1                    VXNORM(195,2), VYNORM(195,2), VZNORM(195,2)
      COMMON/CPRINT/      IPRG  ,
     1                    IPRVES,   IPRVEL,   IPRGAM,   IPRCP ,   IPRCF
C          DIMENSION OF  GAMMAV  IN ACOEFF MUST AGREE WITH  GAMMAV BELOW
      DIMENSION
     1    HDIAG (156,2), HINV     (156), H         (156), G      (156,3)
      DIMENSION
     1    ROOTV (156,2,5),
     2    GAMMAV(156,2,5),
     3    HGAMMA(156,2,5),
     4    VX  (195,2,5), VY  (195,2,5), VZ  (195,2,5)
      DIMENSION
     1    MTIP(3)       , MROOT(3)        ,
     2    IVEL(4)       ,
     3    XP(10)        , YP(8)           , ZP(10)         ,
     4    DOWNV(10,8,10),SIDEV(10,8,10),VMAG(10,8,10),
     5    A(3,3)        ,
     6    UCG(LM) , VCG(LM) , WCG(LM) , P(LM)   , Q(LM)   , R(LM)    ,
     6    X(LM)   , Y(LM)   , Z(LM)   , ANU(LM) , PSI(LM) , OMEGA(LM),
     7    CDTOT(5), CYTOT(5), CLTOT(5), CMTOT(5), CNTOT(5), CRTOT(5)
      EQUIVALENCE      (  ROOTV (1,1,1), HGAMMA(1,1,1)  )
 7100 FORMAT( 16I5    )
 7200 FORMAT( 8F10.0 )
      IF( M-1 )  4000, 4000, 4150
 4000 READ(51)
     1    AMACH           ,
     2    IHGAMA          , ICF             ,
     3    NWP             ,
     4    NCHORD          , NSPAN           ,
     5    NSCORD          ,
     6    MTIP            , MROOT           ,
     7    XCG             , YCG             , ZCG             ,
     8    XM        , ZM        , YL        , ZL        , XN        , YN        ,
     9    S               , BSPAN           , CBAR            ,
     A    NSYM            , IBC             ,
     A    IVEL            ,
     B    IPRG    ,    IPRVES,  IPRVEL,    IPRGAM,    IPRCP ,    IPRCF ,
     D    IX1       , IX2       , IY1       , IY2       , IZ1       , IZ2       ,
     E    XP              , YP              , ZP              ,
     F    DOWNV           , SIDEV           , VMAG
      READ(51)                                                  NVOR        ,
     1    NUC             , NUS             , NVEL            , NUT         ,
     2    XNET            , YNET            , ZNET            ,
     3    AKSI            , ETA             , ZETA            ,
     4    BDC
      REWIND 51
      BETA             = SQRT( 1.-AMACH**2 )
      NVORT            = NVOR(NWP)
 4150 CONTINUE
      READ(21)           HDIAG
C
C
      IKJ  = IKJ + 1
      NIWP = NWP
      DO 6900        ISOLVE=1,4
      GO TO( 5180, 5110, 5110, 5114), ISOLVE
 5110 IF( IVEL(ISOLVE) )  6900, 6900, 5180
 5114 IF( IVEL(4)      )  6900, 6900, 5120
 5120 CONTINUE
C     THE FOLLOWING STATEMENTS, THROUGH 5172, ARE USED TO COMPUTE VELOCI
C     TY AT POINTS OFF THE SURFACE OF THE AERODYNAMIC PLANFORM
      READ(5,7100)         NIWP
      DO 5150  IW=1,NIWP
      READ(5,7100)         NUC(IW,4), NUS(IW,4)
      NVELC          = NUC(IW,4)
      NVELS          = NUS(IW,4)
      DO 7710  II=1,NVELC
 7710 READ(5,7200)        (XFLOWF(II,JJ), JJ=1,NVELS)
      DO 7720  II=1,NVELC
 7720 READ(5,7200)        (YFLOWF(II,JJ), JJ=1,NVELS)
      DO 7730  II=1,NVELC
 7730 READ(5,7200)        (ZFLOWF(II,JJ), JJ=1,NVELS)
 5150 CONTINUE
C
 5160 NVEL(1,4)   = NUC(1,4) * NUS(1,4)
      IF( NIWP-1 )  5172, 5172, 5164
 5164 DO 5170  IW=2,NIWP
 5170 NVEL(IW,4) = NVEL(IW-1,4)  +  NUC(IW,4) * NUS(IW,4)
 5172 NUT(4)          = NVEL(NIWP,4)
C
 5180 NVELT           = NUT(ISOLVE)
      DO 5700           ICASE=1,NCASE
          MM            = 2*K-2+ICASE
          ANU(MM)       = ANU(MM) * BETA
          PSI(MM)       = PSI(MM) * BETA
          CALL     AXES  ( LM, MM,  A,    ANU       , PSI       , OMEGA    )
          ANU(MM)       = ANU(MM) / BETA
          PSI(MM)       = PSI(MM) / BETA
          IF( IBC )  5300, 5300, 5400
 5300     DO 5310        ISYM=1,NSYM
          DO 5310          IT=1,NVELT
          VXNORM(IT,ISYM)= - A(1,1)
          VYNORM(IT,ISYM)= - A(2,1)
 5310     VZNORM(IT,ISYM)= - A(3,1)
          GO TO 5500
 5400     CONTINUE
          XORIG        = X(MM)              * 12. / SCALE
          YORIG        = Y(MM) * BETA * 12. / SCALE
          ZORIG        = Z(MM) * BETA * 12. / SCALE
          CALL     FRESTR( XM, ZM, YL, ZL, XN, YN,
     1    LM              ,
     1    IX1, IX2, IY1, IY2, IZ1, IZ2,
     2    XP              , YP              , ZP              ,
     3    DOWNV           , SIDEV           , VMAG            ,
     4    ISOLVE          , ICASE           , NSYM            ,
     5    XORIG          , YORIG         , ZORIG         ,
     6    A              ,
     7    MH             , U1            , SCALE         ,
     8    UCG      , VCG       , WCG     , P        , Q        , R        )
 5500     DO 5510          ISYM=I,NSYM
          DO 5510            IT=1,NVELT
          VX(IT,ISYM,ICASE) = VXNORM(IT,ISYM)
          VY(IT,ISYM,ICASE) = VYNORM(IT,ISYM)
 5510     VZ(IT,ISYM,ICASE) = VZNORM(IT,ISYM)
 5700 CONTINUE
C
      IF( ISOLVE-I )  6100, 6100, 6400
 6100 DO 6150 ICASE=1,NCASE
      DO 6150  ISYM=1,NSYM
      DO 6150    IT=1,NVORT
      BDOTV(IT,ISYM,ICASE) =   ( - BDC(IT,I) * VX(IT,ISYM,ICASE)
     1                             - BOC(IT,2) * VY(IT,ISYM,ICASE)
     2                             - BOC(IT,3) * VZ(IT,ISYM,ICASE)   )
     3                       / HDIAG(IT,ISYM)
      GO TO( 6150, 6140), NSYM
 6140 BDC(IT,2)      = - BDC(IT,2)
 6150 CONTINUE
C
      DO 6210      ICASE=1,NCASE
      DO 6210        ISYM=I,NSYM
      DO 6210          IT=I,NVORT
 6210 GAMMAV(IT,ISYM,ICASE) = 0.
C
      DO 6250        ISYM=I,NSYM
      DO 6250          IT=1,NVORT
      READ(21)           HINV
          DO 6240 ICASE=1,NCASE
          DO 6240     JT=1,NVORT
          GAMMAV(IT,1,ICASE) =    GAMMAV(IT,1,ICASE)
     1                          + HINV(JT) * BOOTV(JT,ISYM,ICASE)
          GO TO( 6230, 6232), NSYM
 6230     GAMMAV(IT,2,ICASE) = - GAMMAV(IT,1,ICASE)
          GO TO 6240
 6232     GAMMAV(IT,2,ICASE) =    GAMMAV(IT,2,ICASE)
     I                          + HINV(JT) * BOOTV(JT,3-ISYM,ICASE)
 6240     CONTINUE
 6250 CONTINUE
C
      IF( IPRGAM )  8838, 8838, 8828
 8828 WRITE(6,8830)       (ICASE, ICASE=I,NCASE)
 8830 FORMAT('I',2IX,' GAMMAV(IT,ISYM,ICASE)'/ I6X, 5(6X,' ICASE=',I1) )
      DO 8834        ISYM=I,NSYM
      DO 8834          IT=1,NVORT
 8834 WRITE(6,8836)        ISYM, IT, (GAMMAV(IT,ISYM,ICASE), ICASE=I,NCASE)
 8836 FORMAT(                 ' ISYM=',I1,      '    IT=',I3,  8FI4.5          )
 8838 CONTINUE
C
      IF( IHGAMA )  6390, 6390, 6300
 6300 CONTINUE
      WRITE(6,8810)       (ICASE, ICASE=1,NCASE)
 8810 FORMAT('1',21X,' BOOTV(IT,ISYM,ICASE) '/ I6X, 5(6X,' ICASE=',I1) )
      DO 8814  ISYM=1,NSYM
      DO 8814    IT=I,NVORT
 8814 WRITE(6,8816)        ISYM, IT, (ROOTV(IT,ISYM,ICASE), ICASE=1,NCASE)
 8816 FORMAT(                 ' ISYM=',II,   -  '    IT=',I3,  8F14.5          )
C
      DO 6310      ICASE=1,NCASE
      DO 6310        ISYM=I,NSYM
      DO 6310          IT=1,NVORT
 6310 HGAMMA(IT,ISYM,ICASE) = 0.
C
      DO 6350        ISYM=1,NSYM
      DO 6350          IT=I,NVORT
      READ(2I)           H
          DO 6340 ICASE=1,NCASE
          DO 6340     JT=I,NVORT
          HGAMMA(IT,I,ICASE) =    HGAMMA(IT,1,ICASE)
     1                          + H(JT) * GAMMAV(JT,ISYM,ICASE)
          GO TO( 6340, 6332), NSYM
 6332     HGAMMA(IT,2,ICASE) =    HGAMMA(IT,2,ICASE)
     1                          + H(JT) * GAMMAV(JT,3-ISYM,ICASE(
 6340     CONTINUE
 6350 CONTINUE
C
      WRITE(6,8820)       (ICASE, ICASE=1,NCASE)
 8820 FORMAT('1',21X,' HGAMMA(IT,ISYM,ICASE)'/ 16X, 5(6X,' ICASE=',I1) )
      DO 8824        ISYM=I,NSYM
      DO 8824          IT=1,NVORT
 8824 WRITE(6,8826)        ISYM, IT, (HGAMMA(IT,ISYM,ICASE), ICASE=1,NCASE)
 8826 FORMAT(                 ' ISYM=',II,      '    IT=',I3,  8F14.5          )
 6390 IF( IVEL(I) )  6900, 6900, 6400
C
 6400 CONTINUE
      DO 6450           ISYM=1,NSYM
      DO 6450             IT=1,NVELT
      READ(2I)              G
      IF( IPRG      )  8930, 8930, 8912
 8912 IF( IT-I      )  8916, 8916, 8914
 8914 IF( IT-NVELT  )  8930, 8916, 8916
 8916 DO 8920             N=I,3
 8920 WRITE(6,8922)        ISOLVE, ISYM, IT, N,  (G(JT,N), JT=1,NVORT(
 8922 FORMAT(         ' ISOLVE=',I1,  5X, ' ISYM=',I1,  5X, ' IT=',I3,  5X,
     I          ' N=',I1, I5X, ' G(JT,N) READ FROM UNIT(2I)' / (IH 16F8.3))
 8930 CONTINUE
          DO 6440    ICASE=1,NCASE
          DO 6440       JT=1,NVORT
          VX(IT,I,ICASE) =    VX(IT,1,ICASE)
     I                      + G(JT,I) * GAMMAV(JT,ISYM,ICASE)
          VY(IT,I,ICASE) =    VY(IT,I,ICASE)
     1                      + G(JT,2) * GAMMAV(JT,ISYM,ICASE)
          VZ(IT,I,ICASE) =    VZ(IT,I,ICASE)
     1                      + G(JT,3) * GAMMAV(JT,ISYM,ICASE)
          GO TO( 6440, 6432), NSYM
 6432     VX(IT,2,ICASE) =    VX(IT,2,ICASE)
     1                      - G(JT,I) * GAMMAV(JT,3-ISYM,ICASE)
          VY(IT,2,ICASE) =    VY(IT,2,ICASE)
     1                      + G(JT,2) * GAMMAV(JT,3-ISYM,ICASE)
          VZ(IT,2,ICASE) =    VZ(IT,2,ICASE)
     I                      - G(JT,3) * GAMMAV(JT,3-ISYM,ICASE)
 6440     CONTINUE
 6450 CONTINUE
C
C
      DO 6800      ICASE=1,NCASE
      DO 6800        ISYM=I,NSYM
      DO 6580     IW=I,NWP
      NVELC          = NUC(IW,ISOLVE)
      NVELS          = NUS(IW,ISOLVF)
      DO 6580     II=1,NVELC
      DO 6580     JJ=I,NVELS
      IF( IW-1 )  6562, 6562, 6564
 6562 IT             =                       (II-1)*NVELS  +  JJ
      GO TO 6572
 6564 IT             = NVEL(IW-I,ISOLVE)  +  (II-I)*NVELS  +  JJ
 6572 U(II,JJ,IW)    = VX(IT,ISYM,ICASF)
      V(II,JJ,IW)    = VY(IT,ISYM,ICASE)
      W(II,JJ,IW)    = VZ(IT,ISYM,ICASE)
 6580 CONTINUE
C
      IF( IPRVEL )  6694, 6694, 6692
 6692 CALL     VELOCY( ISOLVE, ICASE, NSYM, ISYM )
 6694 MM             = 2*K-2+ICASE
      ALPHA          = ANU(MM) * 180. / 3.1416
      PSIDEG         = PSI(MM) * 180. / 3.1416
      PHI            = OMEGA(MM) * 180. / 3.1416
      CALL     ACOEFF( ISOLVE,  NCASE ,   ICASF ,   NSYM  ,   ISYM  ,
     I LM             ,
     1 IKJ            , ICF           ,
     I MTIP           , MROOT         ,
     1 MM             ,
     2 BETA           , ANU           , PSI           ,
     2 XCG            , YCG           , ZCG           ,
     3 S              , BSPAN         , CBAR          ,
     4 GAMMAV         ,
     C CDTOT    , CYTOT    , CLTOT    , CMTOT    , CNTOT    , CRTOT    )
C
      GO TO( 6800, 6710), NSYM
 6710 DO 6750         IW=I,NIWP
      NNC               = NCHORD(IW)
      NNS               = NSPAN(IW)
      NVORC             = NCHORD(IW)-I
      NVORS      .      = NSPAN(IW) -I
      DO 6720         I=I,NNC
      DO 6720         J=1,NNS
 6720 YNET(I,J,IW)      = - YNET(I,J,IW)
      GO TO( 6730, 6750, 6750, 6750), ISOLVE
 6730 DO 6740         IP=I,NVORC
      DO 6740         IQ=I,NVORS
      ETA(IP,IQ,IW)     = - ETA(IP,IQ,IW)
      IF( IW-I )  6732, 6732, 6734
 6732 IT                =                  (IP-1)*NVORS + IQ
      GO TO 6736
 6734 IT                = NVOR(IW-1) + (IP-1)*NVORS + 10
 6736 BDC(IT,2)         = - BDC(IT,2)
 6740 CONTINUE
 6750 CONTINUE
 6800 CONTINUE
C
 6900 CONTINUE
      REWIND 21
      RETURN
      END
      SUBROUTINE FRESTR( XM, ZM, YL, ZL, XN, YN,
     1    LM                 ,
     1    IX1, IX2, IY1, IY2, IZ1, IZ2,
     2    XP             , YP             , ZP             ,
     3    DOWNV          , SIDEV          , VMAG           ,
     4    ISOLVE         , ICASE          , NSYM           ,
     5    XORIG          , YORIG          , ZORIG          ,
     6    A              ,
     7    HM             , III            , SCALE          ,
     8    UCG      , VCG      , WCG       , P        , Q        , R          )
      COMMON/CNET  /       NWP            , NIWP           ,
     1    NCHORD(3)      , NSPAN (3)      , NSCORD(3)      , NVOR   (3)    ,
     2    NUC   (3,4)    , NUS   (3,4)    , NVEL  (3,4)    , NUT       (4)
      COMMON/CCOORD/       XNET (24,5,3), YNET (24,5,3), ZNET (24,5,3),
     1                     AKSI (23,4,3), ETA  (23,4,3), ZETA (23,4,3),
     2                     XFLOWF( 1, 1), YFLOWF( 1, 1), ZFLOWF( 1, 1),
     3    BOC   (156,3), AX           (3), AY           (3), AZ           (3)
      COMMON/CVELOC/       U     (24,5,3), V     (24,5,3), W     (24,5,3),
     1                     VXNORM(195,2), VYNORM(195,2), VZNORM(195,2)
      COMMON/CPRINT/       IPRG  ,
     1                     IPRVES,    IPRVEL,    IPRGAM,    IPRCP ,    IPRCF
      DIMENSION
     1    XP(10)         , YP(8)          , ZP(10)         ,
     2    DOWNV(10,8,10),SIDEV(10,8,10),VMAG(10,8,10),
     6    A(3,3)         ,
     6    UCG(LM) , VCG(LM) , WCG(LM) , P(LM)    , Q(LM)    , R(LM)    ,
     5    FDOWNV(2)      , FSIDEV(2)      , FVMAG(2)       ,
     6    VPNORM(24,5,3)
C
C
C
      DO 6000   ISYM=1,NSYM
      DO 5950     IW=1,NIWP
      NVELC           = NUC(IW,ISOLVE)
      NVELS           = NUS(IW,ISOLVE)
      DO 5950      II=1,NVELC
      DO 5950      JJ=1,NVELS
      IF( IW-1 )   5502, 5502, 5504
 5502 IT              =                          (II-1)*NVELS + JJ
      GO TO 5506
 5504 IT              = NVEL(IW-1,ISOLVE) + (II-1)*NVELS + JJ
 5506 GO TO( 5510, 5520, 5530, 5540), ISOLVE
 5510 IP              = II
      IQ              = JJ
      PX              = AKSI(IP,IQ,IW)
      PY              =  ETA(IP,IQ,IW)
      PZ              = ZETA(IP,IQ,IW)
      GO TO 5542
C
 5520 I               = II
      J               = JJ
      PX              = ( XNET(I,J,IW) + XNET(I,J+1,IW) ) / 2.
      PY              = ( YNET(I,J,IW) + YNET(I,J+1,IW) ) / 2.
      PZ              = ( ZNET(I,J,IW) + ZNET(I,J+1,IW) ) / 2.
      GO TO 5542
C
 5530 I               = II
      J               = JJ
      PX              = ( XNET(I,J,IW) + XNET(I+1,J,IW) ) / 2.
      PY              = ( YNET(I,J,IW) + YNET(I+1,J,IW) ) / 2.
      PZ              = ( ZNET(I,J,IW) + ZNET(I+1,J,IW) ) / 2.
      GO TO 5542
C
 5540 PX              = XFLOWF(II,JJ)
      PY              = YFLOWF(II,JJ)
      PZ              = ZFLOWF(II,JJ)
C
 5542 PX              = PX - XM
      PY              = PY - YN
      PZ              = PZ - ZL
C
      GO TO( 5546, 5544), ISYM
 5544 PY              = -PY
 5546 CONTINUE
      XVP             = A(1,1) * PX  +  A(2,1) * PY  +  A(3,1) * PZ
      YVP             = A(1,2) * PX  +  A(2,2) * PY  +  A(3,2) * PZ
      ZVP             = A(1,3) * PX  +  A(2,3) * PY  +  A(3,3) * PZ
      XVP             = XVP + XORIG
      YVP             = YVP + YORIG
      ZVP             = ZVP + ZORIG
         IF( XVP - XP(IX1) )  5930, 5610, 5620
 5610    XWT          = ( XVP-XP(IX1) )   /  ( XP(IX1+1)-XP(IX1) )
         IX           = IX1+1
         GO TO 5700
 5620    DO 5632 IX=IX1,IX2
         IF( XVP - XP(IX)   )  5630, 5630, 5632
 5630    XWT          = ( XVP-XP(IX-1) )  /  ( XP(IX)-XP(IX-1) )
         GO TO 5700
 5632    CONTINUE
         GO TO 5930
C
 5700    IF( YVP - YP(IY1) )  5930, 5710, 5720
 5710    YWT          = ( YVP-YP(IY1) )   /  ( YP(IY1+1)-YP(IY1) )
         IY           = IY1+1
         GO TO 5800
 5720    DO 5732 IY=IY1,IY2
         IF( YVP - YP(IY)   )  5730, 5730, 5732
 5730    YWT          = ( YVP-YP(IY-1) )  /  ( YP(IY)-YP(IY-1) )
         GO TO 5800
 5732    CONTINUE
         GO TO 5930
C
 5800    IF( ZVP - ZP(IZ1) )  5930, 5810, 5820
 5810    ZWT          = ( ZVP-ZP(IZ1) )   /  ( ZP(IZ1+1)-ZP(IZ1) )
         IZ           = IZ1+1
         GO TO 5900
 5820    DO 5832 IZ=IZ1,IZ2
         IF( ZVP - ZP(IZ)   )  5830, 5830, 5832
 5830    ZWT          = ( ZVP-ZP(IZ-1) )  /  ( ZP(IZ)-ZP(IZ-1) )
         GO TO 5900
 5832    CONTINUE
         GO TO 5930
C
C        REF. NBS AMS 55, P.882, 25.2.66
 5900    DO 5910 NX=1,2
         FDOWNV(NX)=   (1.-YWT) * (1.-ZWT) * DOWNV(IX-2+NX,IY-1,IZ-1)
     1               +     YWT  * (1.-ZWT) * DOWNV(IX-2+NX,IY  ,IZ-1)
     2               + (1.-YWT) *      ZWT * DOWNV(IX-2+NX,IY-1,IZ  )
     3               +     YWT  *      ZWT * DOWNV(IX-2+NX,IY  ,IZ  )
         FSIDEV(NX)=   (1.-YWT) * (1.-ZWT) * SIDEV(IX-2+NX,IY-1,IZ-1)
     1                  +     YWT   * (1.-ZWT) * S)DEV(IX-2+NX,IY  ,IZ-1)
     2                  + )1.-YWT) *       ZWT  * SIDEV(IX-2+NX,)Y-1,IZ  )
     3                  +     YWT  *       ZWT  * S)DFV(IX-2+NX,1Y  ,I7  )
           FVMAG(NX) =   (1.-YWT) = (1.-ZWT) * VMAG (IX-2+NX,IY-1,IZ-1)
     1                  +     YW1  * (1.-ZW1) * VMAG (IX-?+NX,1Y  ,I7-1)
     2                  + (1.-YWT) *       ZWT  * VMAG ()X-2+NX,IY-1,IZ  )
     3                  +     YWT  *       ZWT  * VMAG (1X-2+NX,1Y  ,1Z  )
 5910      CONTINUE
      OOWNVP          = (1.-XWT) * FDOWNV(1)   +  XWT * FDOWNV(2)
      SIDEVP          = (1.-XWT) * FS(OEV(1)   +  XWT * FSIDEV(2)
      VPMAG           = (1.-XWT) * FVMAG(1)    +  XWT * FVMAG(2)
C     POSIT|VE DOWNWASH ANGLE (EPS)LON) = UPWASH
C     POSITIVE SIDEWASH ANGLE ) SIGMA ) = INWASH ON THE PORTSIDE
      TANEPS          = TAN( DOWNVP * 3.1416/180. )
      TANSIG          = TAN( SIDEVP * 3.1416/I80. (
      UP              = - 1. / SORT( 1. + TANEPS**2 + TANSIG**2 )
      VP              = - TANSIG * UP
      WP              =   TANEPS * UP
      UP              = UP * VPMAG
      VP              = VP * VPMAG
      WP              = WP * VPMAG
      GO TO 5940
 5930 UP              = -1.
      VP              =  0.
      WP              =  0.
 5940 CONTINUE
      VXNORM(IT,ISYM) = AI1,1) * UP  +  A(1,2) * VP  +  A(1,3) * WP
      VYNORM(IT,ISYM) = A(2,1) * UP  +  A(2,2) * VP  +  A)2,3) * WP
      VZNORM(IT,ISYM) = A(3,1) * UP  +  A(3,2) = VP  +  A(3,3) * WP
      UFS             = UCG(MM)  +  Q(MM) * PZ * SCALF / 12.
     1                           -  R(MM) * PY * SCALE / 12.
      VFS             = VCG(MM)  +  R(MM) * PX * SCALF / 12.
     1                           -  P(MM) * PZ * SCALE / 12.
      WFS             = WCG(MM)  -  Q(MM) * PX * SCALE / 12.
     I                           +  P(MM) * PY * SCALF / I2.
      VXNORM(IT,ISYM) = VXNORM(IT,ISYM)  -  UFS / U1
      VYNORM(IT,ISYM) = VYNORM(1T,ISYM)  -  VFS / U1
      VZNORM(IT,ISYM) = VZNURM((T,ISYM)  -  WFS / U1
      VPNORM(11,JJ,1W)= SORT(  VXNORM(1T,1SYM]**2 + VYNORMIIT,1SYM]**2
     1                       + VZNORM(1T,ISYM)**2   )
 5950 CONTINUE
C
C
      IF( IPRVFS )  8750, 8750, 8700
 8700 WRITE(6,8704)              ISOLVE,   ICASE,     NSYM,      ISYM
 8704 FORMAT(//  ROX   '         ISOLVE=',I1, '       ICASE=',I2,
     1                '            NSYM=',I1,        ' ISYM=',I1        )
      WRI1E(6,8706)
 8706 FORMAT(/        '         MAGNITUDE OF FREESTREAM VELOCITY VECTOR.
     1RESULTANT OF VXNORM, VYNORM, VZNORM.                  '/
     2                '         VPNORM(1I,JJ,IW)             ')
      DO 8740     IW=1,NIWP
      WRITE(6,8708)       IW
 8708 FORMAT)//           ' IW=   WING PART  =',I1              1
      NVELC           = NUC(IW,ISOLVE)
      NVELS           = NUS(IW,ISOLVE)
      DO 8730     I1=1,NVELC
 8730 WRITF(6,8732)       I1, (VPNORM(I1,JJ,IW), JJ=1,NVELS)
 8732 FORMAT(             ' I)=',I2, 10F10.5                    )
 8740 CONTINUE
 8750 CONTINUE
C
 6000 CONTTNUE
C
C
C
      IF( IPRVFS )  8850, 8850, 8800
 8800 DO 8840   ISYM=1,NSYM
      WRITE(6,8804)              ISOLVE,   ICASE
 8804 FORMAT('I',ROX '           ISOLVE=',I1, '      ICASE=',I2      )
      WRITE(6,8806)
 8806 FORMAT(//       '         NORMALIZED NONUNIFORM FREESTREAM VELOCITY F
     )IELD INDUCED BY PARENT AIRCRAFT AT POINTS ON STORE AT WHICH VELOCI
     2TY IS COMPUTED.' /
     3                '         VXNORM, VYNORM, VZNORM ARE GIVEN IN THE BOM
     4B AXES REFERENCE SYSTEM.'    )
      DO 8840      IW=1,NIWP
      NVELC            = NUC)IW,ISOLVE)
      NVELS            = NUS((W,ISOLVE)
      WRITE(6,8808)       IW
 8808 FORMAT(//           ' IW=  WING PART  =',I1              I
      DO 8840      N=1,3
      WRITE(6,8812)
 8812 FORMAI( / )
      DO 8840      II=1,NVELC
      IF( IW-1 )  8818, 8818, 8820
 8818 IT1    =                         I1I-1)*NVELS + 1
      IT2    =                          I1   *NVELS
      GO TO 8826
 8820 IT1    = NVEL(IW-1,ISOLVE) + I1I-1)*NVELS + 1
      IT?    = NVEL(IW-1,ISOLVE) +  I1   *NVELS
 8826 GO TO( 8828, 8832, 8836), N
 8828 WRITE)6,8830)  ISYM, IW, II,     IVXNORMI1T,ISYMI, 1T=1T1,1T2)
 8830 FORMAT(     '      VXNORM)1T,ISYM)     ISYM=',I1, 5X, ' IW=',I1,
     1          5X,     ' I1=',I2,    5FI1.3     )
      GO TO 8840
 8832 WRITE(6,8834)  ISYM, IW, I1,     (VYNORM(IT,ISYM], IT=1T1,IT2)
 8834 FORMAT(     '      VYNORM(IT,ISYM)     ISYM=',I1, 5X, ' IW=',I1,
     1          5X,     ' I1=',I2,    5E11.3     )
      GO TO 8840
 8836 WR)TE(6,8838)  ISYM, IW, I1,     (VZNORM(IT,ISYM], IT=1T1,IT2)
 8838 FORMAT)     '      VZNORM(IT,ISYM)     ISYM=',I1, 5X, ' IW=',I1,
     1          5X,     ' I1=',)2,    5E11.3     )
 8840 CONTINUE
 8850 CONTINUE
      RETURN
      END
      SUBROUTINE VELOCYI ISOLVE,  ICASE ,   NSYM  ,   ISYM   I
      COMMON/CNET  /     NWP          , NIWP        ,
     I   NCHORO(3)     , NSPAN )3)    , NSCORO[3)     , NVOR  [3)    ,
     2   NUC    (3,4)  , NUS    [3,4) , NVEL  [3,4)   , NUT      (4I
      COMMON/CVELOC/     U      )24,5,3), V      [24,5,3), W     (24,5,3),
     3                   VXNORM(195,2), VYNDRM(195,2), VZNORN)195,2)
      COMMON/CPRINT/     IPRG   ,
     I                   IPRVFS,   IPRVEL,   IPRGAM,   IPRCP ,   IPRCF
      DIMENSION          UMIOS[24,5,3), VM)OS[24,5,3), WMIOS(24,5,3),
     2                   UMIOC(24,5,3), VMIOC[24,5,3), WMIOC(24,5,3)
      EQU)VALENCE      [ U      (1,1,1), UMIDC )I,1,I), UMIOS [I,1,II I,
     1                 [ V      (I,I,I), VMIDC )1,I,1I, VMIOS )1,I,1) I,
     2                 ) W      [1,I,1), WMIOC (1,1,II, WMIOS [I,1,II I
      GO TO( 8100, 2500, 3500, 8100), ISOLVE
 8100 IF( )PRVEL )  1900, 1900, 8102
 8102 CONTINUE
      WRITE[6,8104)            ISOLVE,   ICASE,    NSYM,     ISYM
 8104 FORMATI'1',80X   '       ISOLVE=',II, '      ICASE=',I2,
     1                 '         NSYM=',I1,      ' ISYM=',II       )
      WRITE(6,8106)
 8106 FORMAT(          '       NORMALIZEO VELOCITY AT 80UNOARY POINTS  =
     IFREE STREAM + G*GAMMA   ')
      00 8150   )W=1,NIWP
      NVELC        = NUC)IW,[SOLVE)
      NVELS        = NUS(IW,[SOLVE)
      WRITE[6,8108)      IW
 8108 FORMATI//        ' IW=  WING PART  =',I1         )
      WRITE(6,8110)
 8110 FORMAT(/         '       U(IP,IO,IW)  '    )
      00 8112  IP=I,NVELC
 8112 WRITE)6,8114)      IP, (U(IP,IO,IW),:IO=1,NVELSI
 8114 FORMAT(          ' IP=',I2,  8F15.4               I
C
      WRITF(6,8120)
 8120 FORMAT(/         '       V(IP,IO,IW)  '    )
      DO 8122  IP=1,NVELC
 8122 WRITE[6,8124)      IP, [V[IP,IO,IW), IO=1,NVELS)
 8124 FORMATI          ' IP=',I2,  8F15.4               ]
C
      WRITF(6,8130)
 8130 FORMAT)/         '       W)IP,IO,IWJ  '    )
      DO 8132  IP=1,NVELC
 8132 WRITE(6,8134)      IP, [W[IP,1O,IW), 1O=1,NVELS)
 8134 FORMAT(          ' IP=',I2,  8F15.4               I
C
 8150 CONTINUE
 1900 RETURN
C
 2500 CONTINUE
      DO 2600  IW=1,NWP
      NVELC        = NUC)IW,2)
      NVELS        = NUS()W,2)
      DO 2600   [=1,NVELC
      DO 2600  [O=1,NVELS
      UMIOS(I,IO,IW) = U(I,)O,IW)
      VMIDS(I,1O,IW) = V(1,IQ,[W)
 2600 WMIDS(I,1O,IW) = W(I,IO,IW)
C
      IF( IPRVEL )  2900, 2900, 8202
 8202 CONTINUE
      WRITE(6,8204)            ISOLVE,   ICASE,    NSYM,     ISYM
 8204 FORMATI'1',90X   '       ISOLVE=',[1, '      ICASE=',I2,
     l                 '         NSYM=',I1,      ' ISYM=',II       I
      WRITE(6,8206)
 8206 FORMAT(          '       NORMALIZED VELOCITY AT SPANWISE SEGMENT MIO
     1POINTS  =  FREESTREAM + G*GAMNA  ')
      DO 8250  IW=1,NWP
      NVELC        = NUC(IW,2)
      NVELS        = NUS(IW,2)
      WRITE(6,8208)      IW
 8208 FORMAT(//        ' IW=  W)NG PART  =',I1          )
      WRITE[6,8210)
 8210 FORMAT)/         '       UMIOS(I,IO,IW)            ')
      OO 8212      [=I,NVELC
 8212 WRITE[6,8214)      I,  (UMIOS)I,IO,IW), [O=1,NVELS)
 8214 FORMAT(          ' ) =',I2,   9F15.4    )
      WRITE[6,8220)
 8220 FORMAT(/         '       VMIOS(I,1O,IW)            ')
      DO 8222      [=1,NVELC
 8222 WRITE(6,8224)      |,  [VMIOS(I,IQ,IW), IO=I,NVELS)
 8224 FORMAT(          ' ] =',I2,   8F15,4    I
      WRITF(6,8230)
 8230 FORMAT(/     '   '       WMIDS)),IO,IW)            ')
      OO 8232      ]=1,NVELC
 8232 WRITE(6,8234)      [,  (WMIOS[I,)Q,IW), IO=1,NVELS)
 8234 FORMAT(          ' ] =',I2,   8F15.4    I
 8250 CONTINUE
 2900 RETURN
C
 3500 CONTINUE
      OO 3600  IW=1,NWP
      NVELC        = NUC[IW,3)
      NVELS        = NUS[IW,3)
      DO 3600  1P=1,NVELC
      OO 3600   J=1,NVELS
      UMIOC(IP,J,[W) = U(1P,J,1W)
      VMIDC(IP,J,[W) = V(1P,J,IW)
 3600 WMIDC(IP,J,IW) = W(IP,J,IW)
C
      IF( IPRVEL )  39D0, 3900, 8302
 8302 CONTINUE
      WRITE[6,8304)            ISOLVE,   ICASE,    NSYM,     ISYM
 8304 FORMAT('1',80X   '       ISOLVE=',I1, '      ICASE=',I2,
     1                 '         NSYM=',I1,      ' ISYM=',I1       I
      WRITE[6,8306)
 8306 FORMAT)          '       NORMALIZED VELOCITY AT CHORDWISE SEGMENT MI
     IDPOINTS  =  FREESTREAM + G*GAMMA                   ')
      DO 8350  IW=1,NWP
      NVELC        = NUC(IW,3)
      NVELS        = NUS(IW,3)
      WRITE(6,8308I      IW
 8308 FORMAT[//        ' IW=  WING PART  =',I1          )
      WRITE(6,8310)
 8310 FORMAT(/         '       UMIOC(IP,J,IW)            'I
      OO 8312      [P=1,NVELC
 8312 WRITF)6,8314)      IP, [UMIOC)IP,J,IW),  J=I,NVELS)
 8314 FORMAT(          ' IP=',I2,  8F15.4                )
      WRITE(6,8320)
 8320 FORMAT(/         '       VMIDC[IP,J,IW)            'I
      DO 8322      IP=1,NVELC
 8322 WRITE(6,8324)        IP, (VMIDC(IP,J,IW),   J=1,NVELS)
 8324 FORMAT(            ' IP=',I2,  8F15.4                  )
      WRITE(6,8330)
 8330 FORMAT(/           '       WMIDC(IP,J,IW)              ')
      DO 8332     IP=1,NVELC
 8332 WRITE(6,8334)        IP, (WMIDC(IP,J,IW),   J=1,NVELS)
 8334 FORMAT(            ' IP=',I2,  8F15.4                  )
 8350 CONTINUE
 3900 RETURN
      END
      SURROUTINE ACOEFF( 1SOLVE,  NCASE ,   1CASE .   NSYM  ,   1SYM  ,
     1    LM     -        ,
     1    IKJ             , ICF           ,
     1    MTIP            , MROOT         ,
     1    MM              ,
     2    BFTA            , ANU           , PSI           ,
     2    XCG             , YCG           , ZCG           ,
     3    S               , BSPAN         , CBAR          ,
     4    GAMMAV          ,
     C    COTOT    , CYTOT    , CLTOT     , CMTOT    , CNTOT    , CRTOT    I
      COMMON/CNET   /       NWP           , NIWP          ,
     1    NCHORD(3)       , NSPAN (3)     , NSCORD(3)     , NVOR  I3I     ,
     2    NUC    (3,4)    , NUS    (3,4)  , NVEL   (3,4)  , NUT       (4)
      COMMON/CCOORO/        XNET (24,5,3), YNET (24,5,3), ZNET (24,5,3),
     1                      AKSI (23,4,3), ETA  (23,4,3), ZETA (23,4,3),
     2                      XFLOWF( 1, 1), YFLOWF( 1, 1), ZFLOWF( 1, 1),
     3    BOC    (156,3), AX          (3), AY          (3), AZ          (3)
      COMMON/CVELOC/        U     (24,5,3), V     (24,5,3), W     (24,5,3),
     1                      VXNORM(195,2), VYNORM(195,2), VZNORM(195,2)
      COMMON/CPRINT/        IPRG   ,
     1                      IPRVFS,    IPRVEL,    IPRGAM,    IPRCP ,    IPRCF
      OIMENSION
     1    MTIP(3)         , MROOT(3)      ,
     2                      ANU    (LM)   , PSI    (LMI   ,
     4    GAMMAV(156,2,5),
     6    COTOT(5), CYTOT(5), CLTOT(5), CMTOT(5), CNTOT(5), CRTOT(5I
      DIMENSION
     1    OELV1(3)        , OELV2(3)      ,
     2    DGDLSP(24, 4), OGOLCH(23, 5),
     3    URPUPP(23, 4), V8PUPP(23, 4), WBPUPP(23, 4I,
     4    URPLOW(23, 4), VBPLOW(23, 4), WBPLOW(23, 4),
     5    CPUPP (23, 4), CPLOW (23, 4)
      DIMENSION             UM1DS(24,5,3), VMIO5(24,5,3I, WMIOS(24,5,3),
     2                      UMIOC(24,5,3), VMIOC(24,5,3), WMIDC(24,5,3)
      DIMENSION
     1    GAMMA(24,6,3)
      DIMENSION
     1    COSP  (24, 4), CY5P   (24, 4), CLSP   (24, 4I,
     2    CMSP  (24, 4), CNSP   (24, 4), CR5P   (24, 4I,
     3    CDCH  (23, 5), CYCH   (23, 5), CLCH   (23, 5),
     4    CMCH  (23, 5), CNCH   (23, 5), CRCH   (23, 5)
      OIMEN5ION             OELX   (195,2), DELY   (195,2I, DELZ  (195,2),
     1    A3(195,2,6)
      EQUIVALENCE         ( U      (1,1,1), UMIDC (1,1,1), UMIOS (1,1,1I ),
     1                    ( V      (1,1,1), VMIOC (1,1,1), VMIOS (1,1,1I ),
     2                    ( W      (1,1,1), WMIOC (1,1,1I, WMIOS (1,1,1) I
C     OEFINE GAMMA(K,LG,IW)
C     LG              = L+1
C     MTIP(IW)        = 0 CONTIGUOUS TIP5 ( CLOSEO 8OOY )
C                     = 1 TERMINAL   TIPS ( WING, TAIL, FIN )
C     MROOT(IW)       = 0 CONTIGUOUS ROOTS ( CLOSEO BOOY, WING, HORIZ TAILI
C                     = 1 TERMINAL   ROOTS ( VERT TAIL, FIN )
C
C
C
      DO 1400      IW=1,NWP
      NNC             = NCHORD(IW)
      NNS             = NSPAN(IW)
      NNSP1           = NSPAN(IW) +1
      NVORC           = NCHORD(IW)-1
      NVORS           = NSPAN(IW) -1
          DO 1110  K=1,NVORC
          JT          = (K-1)*NVORS + 1
          GAMMA(K,       1,IW) =     ( 1-MT1P(IW) )
     1                              * ( -GAMMAV(JT,3-ISYM,ICASEI I
          JT          =  K * NVORS
 1110     GAMMA(K,NVORS+2,1W) =      ( 1-MROOT(IW) )
     1                              * ( -GAMMAV(JT,3-ISYM,ICASE) I
C
          OO 1250 LG=2,NNS
          OO 1250  K=1,NVORC
          IF( IW-1 )  1210, 1210, 1220
 1210     JT          =                (K-1)*NVORS + (LG-1I
          GO TO 1250
 1220     JT          = NVOR(IW-1) + (K-1)*NVOR5 + (LG-1)
 1250     GAMMA(K,LG,IW)          = GAMMAV(JT,ISYM,ICASE)
C
          OO 1350 LG=1,NNSP1
 1350     GAMMA(NNC,LG,IW)        = O.
 1400 CONTINUE
C
      IF( IPRGAM )  8150, 8150, 8100
 8100 CONTINUE
      WRITE(6,8104)              ISOLVE,    ICASE,     NSYM,      ISYM
 8104 FORMAT('1',BOX  '          ISOLVE=',I1, '      ICASE=',I2,
     1            '               NSYM=',I1,        ' ISYM=',I1          I
      WRITE(6,8110)
 8110 FORMAT(          '         GAMMA(K,LG,IW)               'I
      OO 8130   IW=1,NWP
      NNC             = NCHORO(IW)
      NNSP1           = NSPAN(IW) + 1
      WRITE(6,8120)          IW
 8120 FORMAT(//          ' IW=  WING PART  =',I1                I
      OO 8130   K=1,NNC
 8130 WRITE(6,8132)          K,  (GAMMA(K,LG,IW), LG=1,NN5P1I
 8132 FORMAT(             ' K =',I2, 10E12.4                     I
 8150 CONTINUE
C
C
      GO TO( 1500, 2000, 3000), 1SOLVE
 1500 CONTINUE
      OO 1900   IW=1,NWP
      NNC             = NCHORO(IW)
      NNS             = NSPAN(IW)
      NVORC           = NCHORO(IW)-1
      NVORS           = NSPAN(IW)-1
C
C     COMPUTE OGOL AT MIOPOINTS OF SPAN SEGMENTS
      DO 1550     I=1,NNC
      IP              = I
      OO 1550   IO=1,NVORS
      J               = IO
      LG              = IO+1
      IF( I-1    )  1510, 1510, 1508
 1508 IF( I-NNC  )  1520, 1530, 1530
 1510 XLE             = ( XNET(1,J,IW) + XNET(1,J+1,IW) ) / 2.
      YLE             = ( YNET(1,J,IW) + YNET(1,J+1,IW) ) / 2.
      ZLE             = ( ZNET(1,J,IW) + ZNET(1,J+1,IW) ) / 2.
      ELASOC          = 2.* SORT(   ( AKSI( 1,IO,IW)-XLE              I**2
     1                            + (  ETA( 1,IO,IW)-YLE              )**2
     2                         + ( ZETA( 1,IQ,IW)-ZLE            )**2   )
      GO TO 1540
C
 1520 ELASOC        =      SORT(  ( AKSI(IP,IQ,IW)-AKSI(IP-1,IQ,IW) )**2
     1                         + (  ETA(IP,IQ,IW)- ETA(IP-1,IQ,IW) )**2
     2                         + ( ZETA(IP,IQ,IW)-ZETA(IP-1,IQ,IW) )**2   )
      GO TO 1540
C
 1530 XTE           =    ( XNET(NNC,J,IW) + XNET(NNC,J+1,IW) ) / 2.
      YTE           =    ( YNET(NNC,J,IW) + YNET(NNC,J+1,IW) ) / 2.
      ZTE           =    ( ZNET(NNC,J,IW) + ZNET(NNC,J+1,IW) ) / 2.
      ELASOC        =  2.*SORT(  ( AKSI(NVORC,IQ,IW)-XTE            )**2
     1                         + (  ETA(NVORC,IQ,IW)-YTE            )**2
     2                         + ( ZETA(NVORC,IQ,IW)-ZTE            )**2   )
C
 1540 DGOLSP(I,IQ) = GAMMA(I,IQ+1,IW) / ELASOC
 1550 CONTINUE
C
C     COMPUTE DGOL AT MIDPOINTS OF CHORD SEGMENTS
      DO 1650  IP=1,NVORC
      I             = IP
      DO 1650   J=1,NNS
      IQ            = J
      LG            = IQ+1
      IF( J-1    )  1610, 1610, 1604
 1604 IF( J-NNS )   1620, 1630, 1630
C
 1610 IF( MTIP(IW) )  1612, 1612, 1614
 1612 ELASOC        = ABS( 2. * ETA(IP,1,IW) )
      GO TO 1640
 1614 XTIP          = ( XNET(I,1,IW) + XNET(I+1,1,IW) ) / 2.
      YTIP          = ( YNET(I,1,IW) + YNET(I+1,1,IW) ) / 2.
      ZTIP          = ( ZNET(I,1,IW) + ZNET(I+1,1,IW) ) / 2.
      ELASOC        = 2.* SORT(   ( AKSI(IP,1 ,IW)-XTIP             )**2
     1                         + (  ETA(IP,1 ,IW)-YTIP             )**2
     2                         + ( ZETA(IP,1 ,IW)-ZTIP             )**2   )
      GO TO 1640
C
 1620 ELASOC        =      SORT(  ( AKSI(IP,IQ,IW)-AKSI(IP,IQ-1,IW) )**2
     1                         + (  ETA(IP,IQ,IW)- ETA(IP,IQ-1,IW) )**2
     2                         + ( ZETA(IP,IQ,IW)-ZETA(IP,IQ-1,IW) )**2   )
      GO TO 1640
C
 1630 IF( MROOT(IW) )   1632, 1632, 1634
 1632 ELASOC        =      ABS( 2. * ETA(IP,NNS-1,IW) )
      GO TO 1640
 1634 XROOT         = ( XNET(I,NNS,IW) + XNET(I+1,NNS,IW) ) / 2.
      YROOT         = ( YNET(I,NNS,IW) + YNET(I+1,NNS,IW) ) / 2.
      ZROOT         = ( ZNET(I,NNS,IW) + ZNET(I+1,NNS,IW) ) / 2.
      ELASOC        = 2.* SORT(   ( XROOT          -AKSI(IP,NNS-1,IW) )**2
     1                         + ( YROOT          - ETA(IP,NNS-1,IW) )**2
     2                         + ( ZROOT          -ZETA(IP,NNS-1,IW) )**2   )
C
 1640     SUMGAM     = 0.
          DO 1642  K=1,I
 1642     SUMGAM     = SUMGAM + GAMMA(K,LG-1,IW) - GAMMA(K,LG,IW)
 1650 DGOLCH(IP,J) = SUMGAM / ELASOC
C
C
      DO 1700       IP=1,NVORC
      I                = IP
      DO 1700       IQ=1,NVORS
      J                = IQ
      DGOLSP(IP,IQ) = 0.5    *  ( DGOLSP(I,IQ) + DGOLSP(I+1,IQ) )
 1700 DGOLCH(IP,IQ) = 0.5    *  ( DGOLCH(IP,J) + DGOLCH(IP,J+1) )
C
C
      DO 1850  IP=1,NVORC
      I             = IP
      DO 1850  IQ=1,NVORS
      J             = IQ
      IF( IW-1 )  1810, 1810, 1820
 1810 IT            =                  (IP-1)*NVORS + IQ
      GO TO 1830
 1820 IT            = NVOR(IW-1) + (IP-1)*NVORS + IQ
 1830 CONTINUF
      DELXSP        =      ( XNET(I,J+1,IW) + XNET(I+1,J+1,IW) ) / 2.
     1                   - ( XNET(I,J  ,IW) + XNET(I+1,J  ,IW) ) / 2.
      DELYSP        =      ( YNET(I,J+1,IW) + YNET(I+1,J+1,IW) ) / 2.
     1                   - ( YNET(I,J  ,IW) + YNET(I+1,J  ,IW) ) / 2.
      DELZSP        =      ( ZNET(I,J+1,IW) + ZNET(I+1,J+1,IW) ) / 2.
     1                   - ( ZNET(I,J  ,IW) + ZNET(I+1,J  ,IW) ) / 2.
      EL            = SORT( DELXSP**2 + DELYSP**2 + DELZSP**2 )
      COEFF         = 0.5     * DGOLSP(IP,IQ) / EL
      DELV1(1)      = COEFF * (  DELYSP * BOC(IT,3)
     1                         - DELZSP * BOC(IT,2)   )
      DELV1(2)      = COEFF * (  DELZSP * BOC(IT,1)
     1                         - DELXSP * BOC(IT,3)   )
      DELV1(3)      = COEFF * (  DELXSP * BOC(IT,2)
     1                         - DELYSP * BOC(IT,1)   )
C
      DELXCH        =      ( XNET(I+1,J,IW) + XNET(I+1,J+1,IW) ) / 2.
     1                   - ( XNET(I  ,J,IW) + XNET(I  ,J+1,IW) ) / 2.
      DELYCH        =      ( YNET(I+1,J,IW) + YNET(I+1,J+1,IW) ) / 2.
     1                   - ( YNET(I  ,J,IW) + YNET(I  ,J+1,IW) ) / 2.
      DELZCH        =      ( ZNET(I+1,J,IW) + ZNET(I+1,J+1,IW) ) / 2.
     1                   - ( ZNET(I  ,J,IW) + ZNET(I  ,J+1,IW) ) / 2.
      EL            = SORT( DELXCH**2 + DELYCH**2 + DELZCH**2 )
      COEFF         = 0.5     * DGOLCH(IP,IQ) / EL
      DELV2(1)      = COEFF * (  DELYCH * BOC(IT,3)
     1                         - DELZCH * BOC(IT,2)   )
      DELV2(2)      = COEFF * (  DELZCH * BOC(IT,1)
     1                         - DELXCH * BOC(IT,3)   )
      DELV2(3)      = COEFF * (  DELXCH * BOC(IT,2)
     1                         - DELYCH * BOC(IT,1)   )
      UBPUPP(IP,IQ) = U(IP,IQ,IW) + DELV1(1) + DELV2(1)
      VBPUPP(IP,IQ) = V(IP,IQ,IW) + DELV1(2) + DELV2(2)
      WBPUPP(IP,IQ) = W(IP,IQ,IW) + DELV1(3) + DELV2(3)
      UBPLOW(IP,IQ) = U(IP,IQ,IW) - DELV1(1) - DELV2(1)
      VBPLOW(IP,IQ) = V(IP,IQ,IW) - DELV1(2) - DELV2(2)
      WBPLOW(IP,IQ) = W(IP,IQ,IW) - DELV1(3) - DELV2(3)
      CPUPP(IP,IQ)
     1         = 1. - ( U(IP,IQ,IW) + DELV1(1) + DELV2(1) )**2
     2              - ( V(IP,IQ,IW) + DELV1(2) + DELV2(2) )**2
     3              - ( W(IP,IQ,IW) + DELV1(3) + DELV2(3) )**2
      CPLOW(IP,IQ)
     1         = 1. - ( U(IP,IQ,IW) - DELV1(1) - DELV2(1) )**2
     2              - ( V(IP,IQ,IW) - DELV1(2) - DELV2(2) )**2
     3              - ( W(IP,IQ,IW) - DELV1(3) - DELV2(3) )**2
      CPUPP(IP,IQ) = CPUPP(IP,IQ) / BETA**2
      CPLOW(IP,IO) = CPLOW(IP,IO) / BETA**2
C
 1850 CONTINUE
C
C
      IF( IPRVEL )  8350, 8350, 8200
 8200 CONTINUE
      WRITE(6,8204)                ISOLVE,   ICASE,    NSYM,     ISYM
 8204 FORMAT(//  80X  '            ISOLVE=',I1, '      ICASE=',I2,
     1                '              NSYM=',I1,     ' ISYM=',I1     )
      WRITE(6,8206)
 8206 FORMAT(/        '       UPPER BOUNDARY POINT VELOCITY  =  FREE STRE
     1AM + G*GAMMA + DISTRIBUTED VORTICITY                       ')
      WRITE(6,8208)     IW
 8208 FORMAT(         ' IW=  WING PART  =',I1                    )
      WRITE(6,8210)
 8210 FORMAT(/        '       U + DELV1(1) + DELV2(1) ')
      DO 8212     IP=1,NVORC
 8212 WRITE(6,8214)     IP, (UBPUPP(IP,IO), IO=1,NVORS)
 8214 FORMAT(         ' IP=',I2,  8F15.4                         )
      WRITE(6,8220)
 8220 FORMAT(/        '       V + DELV1(2) + DELV2(2) ')
      DO 8222     IP=1,NVORC
 8222 WRITE(6,8224)     IP, (VBPUPP(IP,IO), IO=1,NVORS)
 8224 FORMAT(         ' IP=',I2,  8F15.4                         )
      WRITE(6,8230)
 8230 FORMAT(/        '       W + DELV1(3) + DELV2(3) ')
      DO 8232     IP=1,NVORC
 8232 WRITE(6,8234)     IP, (WBPUPP(IP,IO), IO=1,NVORS)
 8234 FORMAT(         ' IP=',I2,  8F15.4                         )
C
      WRITE(6,8306)
 8306 FORMAT(/        '       LOWER BOUNDARY POINT VELOCITY  =  FREE STRE
     1AM + G*GAMMA + DISTRIBUTED VORTICITY                       ')
      WRITE(6,8308)     IW
 8308 FORMAT(         ' IW=  WING PART  =',I1                    )
      WRITE(6,8310)
 8310 FORMAT(/        '       U - DELV1(1) - DELV2(1) ')
      DO 8312     IP=1,NVORC
 8312 WRITE(6,8314)     IP, (UBPLOW(IP,IO), IO=1,NVORS)
 8314 FORMAT(         ' IP=',I2,  8F15.4                         )
      WRITE(6,8320)
 8320 FORMAT(/        '       V - DELV1(2) - DELV2(2) ')
      DO 8322     IP=1,NVORC
 8322 WRITE(6,8324)     IP, (VBPLOW(IP,IO), IO=1,NVORS)
 8324 FORMAT(         ' IP=',I2,  8F15.4                         )
      WRITE(6,8330)
 8330 FORMAT(/        '       W - DELV1(3) - DELV2(3) ')
      DO 8332     IP=1,NVORC
 8332 WRITE(6,8334)     IP, (WBPLOW(IP,IO), IO=1,NVORS)
 8334 FORMAT(         ' IP=',I2,  8F15.4                         )
 8350 CONTINUE
C
C
      IF( IPRCP )  1900, 1900, 8400
 8400 CONTINUE
      WRITE(6,8404)                ISOLVE,   ICASE,    NSYM,     ISYM
 8404 FORMAT(//  80X  '            ISOLVE=',I1, '      ICASE=',I2,
     1                '              NSYM=',I1,     ' ISYM=',I1     )
      WRITE(6,8408)     IW
 8408 FORMAT(//       ' IW=  WING PART  =',I1                    )
      IPLE          = 1 + NCHORD(IW) - NSCORD(IW)
      WRITE(6,8410)
 8410 FORMAT(         '       CPUPP(IP,IO)          CSUBP UPPER SURFACE  ')
      DO 8412  IP=IPLE,NVORC
 8412 WRITE(6,8414)     IP, (CPUPP(IP,IO), IO=1,NVORS)
 8414 FORMAT(         ' IP=',I2,  8F10.4                         )
      WRITE(6,8420)
 8420 FORMAT(/        '       CPLOW(IP,IO)          CSUBP LOWER SURFACE  ')
      DO 8422  IP=IPLE,NVORC
 8422 WRITE(6,8424)     IP, (CPLOW(IP,IO), IO=1,NVORS)
 8424 FORMAT(         ' IP=',I2,  8F10.4                         )
C
 1900 CONTINUE
      RETURN
C
C
C     COMPUTE FORCE AND MOMENT COEFFICIENTS
C     CLSP          IS LIFT       COEFFICIENT AT MIDPOINT OF SPAN SEGMENT
C     CDSP             DRAG
C     CYSP             SIDE FORCE
C     CLCH             LIFT                                  CHORD
C     CDCH             DRAG
C     CYCH             SIDE FORCE
C     CMSP             PITCH                                 SPAN
C     CRSP             ROLL
C     CNSP             YAW
C     CMCH             PITCH                                 CHORD
C     CRCH             ROLL
C     CNCH             YAW
C     CLTOT            LIFT                SUMMED OVER ALL SEGMENTS
C     CDTOT            DRAG
C     CYTOT            SIDE FORCE
C     CMTOT            PITCH
C     CRTOT            ROLL
C     CNTOT            YAW
 2000 CONTINUE
C     COMPUTE FORCES AND MOMENTS AT SPANWISE MIDPOINTS
      GO TO( 2100, 2200), ISYM
 2100 CDTOT(ICASE) = 0.
      CYTOT(ICASE) = 0.
      CLTOT(ICASE) = 0.
      CMTOT(ICASE) = 0.
      CNTOT(ICASE) = 0.
      CRTOT(ICASE) = 0.
 2200 DO 2600       IW=1,NWP
      NNC           = NCHORD(IW)
      NVORS         = NSPAN(IW) -1
      ILE           = 1 + NCHORD(IW) - NSCORD(IW)
      DO 2500       I=ILE,NNC
      DO 2500      IO=1,NVORS
      J             = IO
      LG            = IO + 1
C
      IF( IW-1     )  2420, 2420, 2422
 2420 IT            =                 (I-1)*NVORS + IO
      GO TO 2430
 2422 IT            = NVEL(IW-1,2) + (I-1)*NVORS + IO
 2430 IF( IKJ  -1 )  2432, 2432, 2460
 2432 IF( ICASE-1 )  2434, 2434, 2460
 2434 IF( ISYM -1 )  2436, 2436, 2460
 2436 DELX(IT,1)    = XNET(I,J+1,IW) - XNET(I,J,IW)
      DELY(IT,1)    = YNET(I,J+1,IW) - YNET(I,J,IW)
      DELZ(IT,1)    = ZNET(I,J+1,IW) - ZNET(I,J,IW)
      XMIDS         = ( XNET(I,J,IW) + XNET(I,J+1,IW) ) / 2.
      YMIDS         = ( YNET(I,J,IW) + YNET(I,J+1,IW) ) / 2.
      ZMIDS         = ( ZNET(I,J,IW) + ZNET(I,J+1,IW) ) / 2.
      A3(IT,1,1)    = (2./(S*CBAR))  * (XMIDS-XCG)
      A3(IT,1,2)    = (2./(S*CBAR))  * (ZMIDS-ZCG)
      A3(IT,1,3)    = (2./(S*CBAR))  * (YMIDS-YCG)
      A3(IT,1,4)    = (2./(S*CBAR))  * (XMIDS-XCG)
      A3(IT,1,5)    = (2./(S*BSPAN)) * (YMIDS-YCG)
      A3(IT,1,6)    = (2./(S*BSPAN)) * (ZMIDS-ZCG)
 2460 CONTINUE
C
      PX            = - GAMMA(I,LG,IW)
     1        * ( WMIDS(I,IQ,IW)*DELY(IT,1) - VMIDS(I,IQ,IW)*DELZ(IT,1) )
      PY            = - GAMMA(I,LG,IW)
     1        * ( UMIDS(I,IQ,IW)*DELZ(IT,1) - WMIDS(I,IQ,IW)*DELX(IT,1) )
      PZ            = - GAMMA(I,LG,IW)
     1        * ( VMIDS(I,IQ,IW)*DELX(IT,1) - UMIDS(I,IQ,IW)*DELY(IT,1) )
      COSP(I,IQ)    = - (2./S) * PX
      CYSP(I,IQ)    =   (2./S) * PY
      CLSP(I,IQ)    = - (2./S) * PZ
      CMSP(I,IQ)    = - PZ * A3(IT,1,1) + PX * A3(IT,1,2)
      CNSP(I,IQ)    = - PX * A3(IT,1,3) + PY * A3(IT,1,4)
      CRSP(I,IQ)    =   PZ * A3(IT,1,5) - PY * A3(IT,1,6)
      GO TO( 2472, 2470), NSYM
 2470 DELY(IT,1)    = - DELY(IT,1)
      A3(IT,1,3)    = - A3(IT,1,3)
      A3(IT,1,5)    = - A3(IT,1,5)
 2472 CONTINUE
C
      COTOT(ICASE) = COTOT(ICASE) + COSP(I,IQ)
      CYTOT(ICASE) = CYTOT(ICASE) + CYSP(I,IQ)
      CLTOT(ICASE) = CLTOT(ICASE) + CLSP(I,IQ)
      CMTOT(ICASE) = CMTOT(ICASE) + CMSP(I,IQ)
      CNTOT(ICASE) = CNTOT(ICASE) + CNSP(I,IQ)
      CRTOT(ICASE) = CRTOT(ICASE) + CRSP(I,IQ)
 2500 CONTINUE
C
      IF( IPRCF )  8670, 8670, 8600
 8600 CONTINUE
      WRITE(6,8604)              ISOLVE,   ICASE,    NSYM,      ISYM
 8604 FORMAT(//  BOX   '         ISOLVE=',I1, '      ICASE=',I2,
     1                '            NSYM=',I1,       ' ISYM=',I1      )
      WRITE(6,8608)        IW
 8608 FORMAT(//          ' IW=  WING PART  =',I1        )
      WRITE(6,8610)
 8610 FORMAT(/           '       CLSP(I,IQ)              ')
      DO 8612        I=ILE,NNC
 8612 WRITE(6,8614)        I,  (CLSP(I,IQ), IQ=1,NVORS)
 8614 FORMAT(          ' I =',I2,  4F10.5                )
      WRITE(6,8620)
 8620 FORMAT(/           '       COSP(I,IQ)              ')
      DO 8622        I=ILE,NNC
 8622 WRITE(6,8624)        I,  (COSP(I,IQ), IQ=1,NVORS)
 8624 FORMAT(          ' I =',I2,  4F10.5                )
      WRITE(6,8630)
 8630 FORMAT(/           '       CYSP(I,IQ)              ')
      DO 8632        I=ILE,NNC
 8632 WRITE(6,8634)        I,  (CYSP(I,IQ), IQ=1,NVORS)
 8634 FORMAT(          ' I =',I2,  4F10.5                )
      WRITE(6,8640)
 8640 FORMAT(/           '       CMSP(I,IQ)              ')
      DO 8642        I=ILE,NNC
 8642 WRITE(6,8644)        I,  (CMSP(I,IQ), IQ=1,NVORS)
 8644 FORMAT(          ' I =',I2,  4F10.5                )
      WRITE(6,8650)
 8650 FORMAT(/           '       CRSP(I,IQ)              ')
      DO 8652        I=ILE,NNC
 8652 WRITE(6,8654)        I,  (CRSP(I,IQ), IQ=1,NVORS)
 8654 FORMAT(          ' I =',I2,  4F10.5                )
      WRITE(6,8660)
 8660 FORMAT(/           '       CNSP(I,IQ)              ')
      DO 8662        I=ILE,NNC
 8662 WRITE(6,8664)        I,  (CNSP(I,IQ), IQ=1,NVORS)
 8664 FORMAT(          ' I =',I2,  4F10.5                )
 8670 CONTINUE
 2600 CONTINUE
      RETURN
C
C
 3000 CONTINUE
C     COMPUTE FORCES AND MOMENTS AT CHORDWISE MIDPOINTS
      DO 3600      IW=1,NWP
      NNS   = NSPAN(IW)
      GO TO( 3100, 3200), ISYM
 3100 ISYM = 1
      J1    = 1
      J2    = NNS
      GO TO 3400
C
 3200 ISYM = 2
      IF( MTIP(IW) )  3210, 3210, 3220
 3210 J1    = 2
      GO TO 3300
 3220 J1    = 1
C
 3300 IF( MROOT(IW) )  3310, 3310, 3320
 3310 J2    = NNS-1
      GO TO 3400
 3320 J2    = NNS
C
 3400 IPLE            = 1 + NCHORD(IW) - NSCORD(IW)
      NVORC           = NCHORD(IW)-1
      DO 3500      IP=IPLE,NVORC
      I               = IP
      DO 3500       J=J1,J2
      LG              = J+1
C
      IF( IW-1     )  3420, 3420, 3422
 3420 IT              =                     (IP-1)*NSPAN(IW) + J
      GO TO 3430
 3422 IT              = NVEL(IW-1,3) + (IP-1)*NSPAN(IW) + J
 3430 IF( IKJ  -1 )   3432, 3432, 3460
 3432 IF( ICASE-1 )   3434, 3434, 3460
 3434 IF( ISYM -1 )   3436, 3436, 3460
 3436 DELX(IT,2)      = XNET(I+1,J,IW) - XNET(I,J,IW)
      DELY(IT,2)      = YNET(I+1,J,IW) - YNET(I,J,IW)
      DELZ(IT,2)     = ZNET(I+I,J,IW) - ZNET(I,J,IW)
      XMIDC          = ( XNET(I,J,IW) + XNET(I+1,J,IW( )  /  2.
      YMIDC          = ( YNET(I,J,IW) + YNET(I+1,J,IW) (  /  2.
      ZMIDC          = ( ZNET(I,J,IW) + ZNET(I+1,J,IW) )  /  2.
      A3(IT,2,1)     = (2./(S*CBAR))  * (XMIDC-XCG)
      A3()T,2,2)     = (2./(S*CBAR))  * (ZMIDC-ZCG)
      A3(IT,2,3)     = (2./(S*CBAR))  * (YMIDC-YCG)
      A3(IT,2,4)     = (2./(S*CBAR))  * (XMIDC-XCG)
      A3(IT,2,5)     = (2./(S*BSPAN)) * (YMIDC-YCG)
      A3()T,2,6)     = (2./(S*BSPAN)) * (ZMIDC-ZCG(
 3460 CONTINUE
C
          SUMGAM     = 0.
          DO 3490  K=I,I
 3490     SUMGAM     = SUMGAM + GAMMA(K,LG-),)W) - GAMMA(K,LG,IW)
C
      PX             = - SUMGAM
     I        * ( WMIDC()P,J,IW)*DELY(IT,2) - VMIDC(IP,J,IW)*DELZ(IT,2) )
      PY             = - SUMGAM
     1        * ( UMIDC()P,J,IW)*DELZ()T,2) - WMIDC()P,J,IW)*DELX()T,2( I
      PZ             = - SUMGAM
     I        * ( VMIDC(IP,J,IW)*DELX(IT,2) - UMIDC(IP,J,IW)*DELY(IT,2) (
      CDCH(IP,J)     = - (2./S) * PX
      CYCH(IP,J)     =   (2./S) * PY
      CLCH(IP,J)     = - (2./S) * PZ
      CMCH(IP,J)     = - PZ * A3(IT,2,1)  +  PX * A3(IT,2,2)
      CNCH(IP,J)     = - PX * A3(IT,2,3)  +  PY * A3(IT,2,4)
      CRCH(IP,J)     =   PZ * A3(IT,2,5)  -  PY * A3(IT,2,6(
      GO TO( 3472, 3470), NSYM
 3470 DELY(IT,2)     = - DELY(IT,2)
      A3(IT,2,3)     = - A3(IT,2,3)
      A3(IT,2,5)     = - A3(IT,2,5)
 3472 CONTINUE
C
      CDTOT(ICASE) = CDTOT(ICASE) + CDCH(IP,J)
      CYTOT(ICASE) = CYTOT(ICASE) + CYCH(IP,J(
      CLTOT(ICASE) = CLTOT(ICASE) + CLCH(IP,J(
      CMTOT(ICASE) = CMTOT(ICASE) + CMCH(IP,J)
      CNTOT(ICASE) = CNTOT(ICASE) + CNCH()P,J(
      CRTOT(ICASE) = CRTOT(ICASE) + CRCH(IP,J)
 3500 CONTINUE
C
      IF( IPRCF )  8770, 8770, 8700
 8700 CONTINUE
      WRITE(6,8704)                ISOLVE,    ICASE,     NSYM,      ISYM
 8704 FORMAT(//  BOX   '           ISOLVE=',I1, '         ICASE=',I2,
     1                '              NSYM=',I1,         ' ISYM=',I1      (
      WRITE(6,8708)           IW
 8708 FORMAT(//           '   IW=  WING PART  =',I1    )
      WRITE(6,8710)
 8710 FORMAT(/            '         CLCH(IP,J)         '(
      DO 8712      IP=IPLF,NVORC
 8712 WRITE(6,8714)         IP, (CLCH(IP,J), J=J),J2(
 8714 FORMAT(             ' IP=',I2,  5F10.2           (
      WRITE(6,8720)
 8720 FORMAT(/            '         CDCH(IP,J)         '(
      DO 8722      IP=IPLE,NVORC
 8722 WRITE(6,8724)         IP, (CDCH(IP,J), J=J),J2(
 8724 FORMAT(             ' IP=',I2,  5E10.2           I
      WRITE(6,8730)
 8730 FORMAT(/            '         CYCH(IP,J)         ')
      DO 8732      IP=IPLE,NVORC
 8732 WRITE(6,8734)         IP, (CYCH(IP,J), J=J),J2)
 8734 FORMAT(             ' IP=',I2,  5E10.2           (
      WRITE(6,8740)
 8740 FORMAT(/            '         CMCH(IP,J)         '(
      DO 8742      )P=IPLF,NVORC
 8742 WRITE(6,8744)         IP, (CMCH(IP,J), J=J1,J2)
 8744 FORMAT(             ' IP=',I2,  5E10.2           )
      WRITE(6,8750)
 8750 FORMAT(/            '         CRCH(IP,J)         '(
      DO 8752      )P=IPLF,NVORC
 8752 WRITE(6,8754)         IP, (CRCH(IP,J), J=J1,J2)
 8754 FORMAT(             ' IP=',I2,  5F10.2           (
      WRITE(6,8760(
 8760 FORMAT(/            '         CNCH(IP,J)         ')
      DO 8762      IP=IPLE,NVORC
 8762 WRITE(6,8764)         IP, (CNCH(IP,J), J=J1,J2|
 8764 FORMAT(             ' IP=',I2,  5E10.2           |
 8770 CONTINUE
 3600 CONTINUE
C
      GO TO( 4050, 4020), NSYM
 4020 GO TO( 7000, 4200), ISYM
 4050 CDTOT(ICASE) = 2. * CDTOT(ICASE)
      CYTOT(ICASE) = 0.
      CLTOT(ICASE) = 2. * CLTOT(ICASE)
      CMTOT(ICASE( = 2. * CMTOT(ICASE)
      CNTOT(ICASE) = 0.
      CRTOT(ICASE) = 0.
 4200 CONT)NUE
      AINCID         =   ARCOS(  COS(BETA*ANU(MM)( * COS(BFTA*PSI(MM)(  )
     1                 * 180. / 3.1416
C     THE FOLLOWING 2 STATEMENTS ARE PECULIAR TO THE M-117 BOMB.  THEY
C     PRESCRIBE AXIAL FORCE CORRECTIONS FOR )NTERNAL BASE PRESSURE OF A
C     BODY OF REVOLUTION, AND SKIN FRICTION.
      DELCAB         = .4455  *  (.992 - .00018*AINCID**2(
      CDTOT(ICASE) =    ( CDTOT(ICASF) + DELCAB ( / BETA**2
     I                 + .0316*(2-ICF) + .0935*(ICF-1(
      CYTOT(ICASE) = CYTOT(ICASE) / BETA
      CLTOT(ICASE) = CLTOT(ICASE) / BETA
      CMTOT(ICASE) = CMTOT(ICASE) / BETA
      CNTOT(ICASE) = CNTOT(ICASE) / BETA
      CRTOT(ICASE) = CRTOT(ICASE) / BETA
 7000 RETURN
      END
```



```
      SUBROUTINE AXES  ( LM, MM,  A,    ANU      , PSI      , OMEGA    )
C     THIS SUBROUTINE ASSUMES PITCH-YAW-ROLL ROTATION SEQUENCE
C     REF. THELANDER FDL-TDR-64-70  AD617354  P.23
      DIMENSION          A(3,3)        , ANU(LM) , PSI(LM) , OMEGA(LM)
         COSNU      = COS( ANU(MM) )
         SINNU      = SIN( ANU(MM) )
         COSPSI     = COS( PSI(MM) )
         SINPSI     = SIN( PSI(MM) )
         COSOMG     = COS( OMEGA(MM) )
         SINOMG     = SIN( OMEGA(MM) )
         A(1,1)     =   COSNU * COSPSI
         A(1,2)     =           SINPSI
         A(1,3)     = - SINNU * COSPSI
         A(2,1)     =   SINNU          * SINOMG - COSNU * SINPSI * COSOMG
         A(2,2)     =           COSPSI * COSOMG
         A(2,3)     =   COSNU          * SINOMG + SINNU * SINPSI * COSOMG
         A(3,1)     =   SINNU          * COSOMG + COSNU * SINPSI * SINOMG
         A(3,2)     = -         COSPSI * SINOMG
         A(3,3)     =   COSNU          * COSOMG - SINNU * SINPSI * SINOMG
      RETURN
      END
```

Output Data

All of the output data from this step is written on a printer. Distributions of NUFF data, velocity, vorticity, pressure coefficient, and force coefficients can be optionally printed (these options are controlled by input data to the previous step.) When force coefficients are determined without performing trajectory calculations, the output data includes total force and moment coefficients and the orientation of the store with respect to its parent aircraft. When trajectories are calculated, total force and moment coefficients and solutions to the equations of motion are printed.

In performing the calculations for the sample run used to illustrate program operation, it was assumed that the downwash and sidewash vanished identically (DOWNV = 0 and SIDEV = 0) and the distribution of velocity magnitude was constant (VMAG = 1.0).

Representative results for this run (pressure coefficient distribution and total force and moment coefficients) are shown on the following page.

```
IW=  WING PART  =1
     CPUPP(IP,IQ)               CSUBP UPPER SURFACE
IP= 5    -0.0709    -0.0840    -0.0942    -0.0214
IP= 6    -0.0969    -0.0595    -0.0998     0.0251
IP= 7    -0.1334    -0.0342    -0.1254     0.0698
IP= 8    -0.1124    -0.0106    -0.1225     0.0996
IP= 9    -0.0351    -0.0581    -0.0834     0.0386
IP=10     0.0707    -0.0027    -0.0130     0.0592
IP=11     0.1209     0.0273    -0.0074     0.0388
IP=12     0.0648    -0.0327    -0.0789    -0.0463
IP=13     0.0220    -0.0722    -0.1184    -0.0893
IP=14    -0.0708    -0.1564    -0.1942    -0.1622
IP=15    -0.1179    -0.1861    -0.2004    -0.1525
IP=16    -0.0763    -0.1350    -0.1352    -0.0768
IP=17    -0.0904    -0.1420    -0.1330    -0.0686
IP=18    -0.1754    -0.2137    -0.1886    -0.1148
IP=19    -0.2902    -0.3018    -0.2427    -0.1477
IP=20    -0.3503    -0.3258    -0.2197    -0.0942
IP=21    -0.2995    -0.2372    -0.0821     0.0749
IP=22    -0.1908    -0.0903     0.1138     0.3019
IP=23     0.3894     0.4336     0.5740     0.7287

IW=  WING PART  =2
     CPUPP(IP,IQ)               CSUBP UPPER SURFACE
IP= 1    -0.1934    -0.1648    -0.2494     0.2497
IP= 2    -0.0712    -0.0977    -0.0119    -1.1901
IP= 3    -0.0353    -0.0584    -0.0939     0.3197
IP= 4    -0.0253    -0.0218    -0.0543    -1.0630
IP= 5    -0.0243    -0.0254     0.0303     0.3030
IP= 6    -0.0250    -0.0166    -0.1010    -0.6495
IP= 7    -0.0295     0.0058     0.0542     0.2533
IP= 8    -0.0319    -0.0035    -0.0171    -0.3070

     CPLOW(IP,IQ)               CSUBP LOWER SURFACE
IP= 1     0.0933     0.1358     0.0621     0.4734
IP= 2     0.0315     0.0679     0.1525    -0.9469
IP= 3    -0.0040     0.0469     0.0182     0.4215
IP= 4    -0.0318     0.0380     0.0440    -0.6935
IP= 5    -0.0534     0.0029     0.0786     0.2912
IP= 6    -0.0710    -0.0053    -0.0629    -0.1602
IP= 7    -0.0895    -0.0028     0.0748     0.0682
IP= 8    -0.1056    -0.0314    -0.0379     0.1419

IW=  WING PART  =3
     CPUPP(IP,IQ)               CSUBP UPPER SURFACE
IP= 1    -0.2205    -0.1740    -0.2404     0.2740
IP= 2    -0.0964    -0.0852     0.0040    -1.0834
IP= 3    -0.0642    -0.0535    -0.0597     0.3637
IP= 4    -0.0626    -0.0703    -0.0410    -0.9061
IP= 5    -0.0723    -0.0290     0.0498     0.3426
IP= 6    -0.0840    -0.0305    -0.0917    -0.5164
IP= 7    -0.0972    -0.0148     0.0485     0.2816
IP= 8    -0.1081    -0.0294    -0.0165    -0.2056

     CPLOW(IP,IQ)               CSUBP LOWER SURFACE
IP= 1     0.1861     0.2154     0.1360     0.5103
IP= 2     0.1077     0.1313     0.2058    -0.9665
IP= 3     0.0714     0.1019     0.0831     0.4234
IP= 4     0.0464     0.0913     0.0869    -0.5996
IP= 5     0.0284     0.0557     0.1273     0.2645
IP= 6     0.0146     0.0430    -0.0219    -0.1129
IP= 7    -0.0007     0.0444     0.1041     0.0119
IP= 8    -0.0166     0.0137    -0.0008     0.1484
```

```
IRUN= 1   XORIG(IN)=  0.0   YORIG(IN)=  0.0   ZORIG(IN)=  0.0   PITCH(DEG)=  10.0   YAW(DEG)=   0.0   ROLL(DEG)=   0.0
          FORCE COEFFICIENTS FROM  LAMINAR  SKIN FRICTION ASSUMPTION
          FORCE COEFFICIENTS IN WIND AXES SYSTEM
          CD(DRAG)        CY(SIDE FORCE) CL(LIFT)       CR(ROLL)        CM(PITCH)       CN(YAW)
IRUN= 1   0.12732         0.00009        0.60243        -0.00001        -0.15739        -0.00000
          FORCE COEFFICIENTS IN BODY AXES SYSTEM
          CAXIAL          CSIDE          CNORM          CROLL           CPITCH          CYAW
IRUN= 1   0.02074         0.00009        0.61558        -0.00000        -0.15739        -0.00000
```

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1 ORIGINATING ACTIVITY *(Corporate author)* Arnold Engineering Development Center Arnold Air Force Station, Tennessee 37389	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b GROUP N/A

3 REPORT TITLE

AERODYNAMIC FORCES AND TRAJECTORIES OF SEPARATED STORES IN DISTURBED FLOW FIELDS

4 DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

August 16, 1971, through June 30, 1972 -- Final Report

5 AUTHOR(S) *(First name, middle initial, last name)*

W. N. MacDermott and P. W. Johnson, ARO, Inc.

6 REPORT DATE March 1973	7a. TOTAL NO OF PAGES 103	7b. NO OF REFS 6

8a CONTRACT OR GRANT NO b. PROJECT NO c. Program Element 62602F d	9a. ORIGINATOR'S REPORT NUMBER(S) AEDC-TR-72-162 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)* ARO-PWT-TR-72-148

10 DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11 SUPPLEMENTARY NOTES Available in DDC.	12 SPONSORING MILITARY ACTIVITY Air Force Armament Laboratory, Air Force Systems Command, Eglin AFB, FL 32542

13 ABSTRACT

A vortex-lattice potential flow computer program capable of accepting nonuniform flow boundary conditions but previously restricted to incompressible flows with symmetry was modified to eliminate these restrictions. The program was structured in such a way that, after preliminary calculations of a purely geometric nature were performed one time for a given body, potential flow solutions for any set of boundary conditions on that body could be obtained in computer times measured in seconds rather than minutes. The aerodynamic characteristics of an M-117 bomb, represented by a network of 312 vortices, were calculated for uniform flow at a Mach number of 0.5 and were found to agree with wind tunnel measurements to within 10 percent, except for drag. The program was also used to compute forces on an M-117 bomb at a number of different locations in the disturbed flow field generated by an F-4C parent aircraft. In this case, absolute values of the force coefficients were generally in poor agreement with wind tunnel values, but the incremental variations of the calculated coefficients through the nonuniform flow field were within the range from 5 to 10 percent of wind tunnel measurements. A store separation routine was added to the potential flow program, and several representative store separation trajectories were calculated.

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
external stores						
vortex flow						
M-117 bomb						
F-4C aircraft						
wind tunnels						
uniform flow						
nonuniform flow						
grids						
potential flow						
computer programs						
mathematical models						
separation						

AFSC
Arnold AFS Tenn